AD 740788

Technical Report: NAVTRADEVCEN 70-C-0262-2

DEVELOPMENT OF A HYBRID RADAR LANDMASS SIMULATOR: ENGINEERING REPORT 7

Paul A. T. Wolfgang

PENNSYLVANIA RESEARCH ASSOCIATES, INC.
101 North 33 Street
Philadelphia, Pennsylvania 19104

Contract N61339-70-C-0262
NAVTRADEVCEN Task No. 7884

June 1971

DoD Distribution Statement

Approved for public release; distribution unlimited.

NAVAL TRAINING DEVICE CENTER

ORLANDO, FLORIDA

SEE AD 740411

Unclassified

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)

PENNSYLVANIA RESEARCH ASSOCIATES INC.
101 North 33 Street
Philadelphia, Pennsylvania 19104

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
NA

3. REPORT TITLE

DEVELOPMENT OF A HYBRID RADAR LANDMASS SIMULATOR: ENGINEERING REPORT 7

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
Engineering Report covering contract period June 1970 to June 1971

5. AUTHOR(S) (First name, middle initial, last name)

Paul A. T. Wolfgang

6. REPORT DATE
June 1971

7a. TOTAL NO. OF PAGES
137 + vii

7b. NO. OF REFS
12

8a. CONTRACT OR GRANT NO.
N61339-70-C-0262

b. PROJECT NO.

c.

d.

9a. ORIGINATOR'S REPORT NUMBER(S)
U71-1298

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
NAVTRADEVCEN 70-C-0262-2

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES
None

12. SPONSORING MILITARY ACTIVITY
U. S. Naval Training Device Center
Orlando, Florida 32812

13. ABSTRACT

An integrated set of computer software is described for the generation of airborne radar displays, simulating the earth's topographic features and man-made objects to the extent they are apparent on the indicator of a modern scanning radar, as a function of the aspect from which the earth is viewed and the salient characteristics of the viewing radar system. The programs are based on previous PRA work in the simulation of air-to-ground radar displays and provide a versatile laboratory tool for the evaluation of radar simulation techniques with a variety of user options for adjusting scale factors, inserting/deleting special effects, changing the precision of the calculations, incorporating nonlinearities, and varying characteristics of the simulated radar and its antenna. Portions of the software have been developed to enable experimentation on a general-purpose digital computer, with the design parameters (e.g., word length, accuracy, and transfer functions) of the digital and analog processing that will later be cast in special-purpose hardware for a training device. Other portions of the software represent programs that will actually be executed to prepare the radar simulation data base and to generate the display in real time. The set of software described herein is designed to operate in non-real time, although it develops an output file that can be presented to an observer (via a special-purpose display device) in the time frame at which a high-resolution scanning radar in a Mach 2 aircraft views its ground targets.

DD FORM 1 NOV 65 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

Unclassified

Security Classification

Unclassified
Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Radar Landmass Simulator						
Culture Data Displays						
Terrain Data Displays						
Digital Computer						
Computer Programming						
Multiprogramming						

Unclassified
Security Classification

Technical Report: NAVTRADEVCEN 70-C-0262-2

DEVELOPMENT OF A HYBRID RADAR LANDMASS SIMULATOR: ENGINEERING REPORT 7

ABSTRACT

An integrated set of computer software is described for the generation of airborne radar displays, simulating the earth's topographic features and man-made objects to the extent they are apparent on the indicator of a modern scanning radar, as a function of the aspect from which the earth is viewed and the salient characteristics of the viewing radar system. The programs are based on previous PRA work in the simulation of air-to-ground radar displays and provide a versatile laboratory tool for the evaluation of radar simulation techniques with a variety of user options for adjusting scale factors, inserting/deleting special effects, changing the precision of the calculations, incorporating nonlinearities, and varying characteristics of the simulated radar and its antenna. Portions of the software have been developed to enable experimentation on a general-purpose digital computer, with the design parameters (e.g., word length, accuracy, and transfer functions) of the digital and analog processing that will later be cast in special-purpose hardware for a training device. Other portions of the software represent programs that will actually be executed to prepare the radar simulation data base and to generate the display in real-time. The set of software described herein is designed to operate in non-real time, although it develops an output file that can be presented to an observer (via a special-purpose display device) in the time frame at which a high-resolution scanning radar in a Mach 2 aircraft views its ground targets.

FOREWORD

Since the development of air-to-ground mapping radar in the 1940's,a need for radar-simulator training devices has existed. This need has been met, until only recently, by analog/optical systems in which aerial photographs are scanned optically to simulate an actual scanning radar. The photographic transparencies, however, are expensive and difficult to update, their resolution has a low practical limit, and the associated hardware is generally unwieldy. The Naval Training Device Center (NAVTRADEVCEN) has long been actively engaged in research into digital radar landmass simulation.

It has long been recognized that the radar data base represented by transparencies could be stored and handled digitally, precluding the problems mentioned above. However, speed and cost constraints of available digital hardware have, in the past, prevented an all-digital approach to radar simulation. Today's hardware capabilities and cost make an all-digital approach feasible.

This report describes software written by Pennsylvania Research Associates, Inc.,for NAVTRADEVCEN which formats and compresses digitally-encoded terrain data for a representative portion of the United States. Software is also described which processes this terrain data along with cultural information to produce intensity profiles for display on NAVTRADEVCEN's special hybrid equipment. The several intensity profiles, when properly combined and displayed, are intended to simulate an analogous radar scan of the area.

It should be pointed out that NAVTRADEVCEN required the contractor to use FORTRAN IV as the main programming language for all software developed under this contract. While the contractor has pointed out that this restriction often leads to long run times for some of his software, it is nevertheless in the best interest of the Government, because FORTRAN IV is a powerful high-level language producing source programs which are concise, easily interpreted and readily modified. Furthermore, the extended version of FORTRAN IV provided to the contractor allows the arbitrary insertion of assembly language statements within the FORTRAN source program wherever desired.

CONCLUSION: The terrain data base developed by the contractor under this contract is a valid compression of the original digital terrain data. While the contractor did not produce simulated radar scans on the display device, NAVTRADEVCEN personnel have used this data base to produce simulated radar scans, demonstrating the technical feasibility of an all-digital simulator.

Alexander J. Grant

Project Engineer
Naval Training Device Center

The findings in this report are not to be construed as an official Department of the Navy position.

TABLE OF CONTENTS cont'd.

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

SECTION I

INTRODUCTION

This is the final report on Contract N61339-70-C-0262 between the U. S. Naval Training Device Center (NTDC) and Pennsylvania Research Associates Inc. (PRA). The purpose of the contract is to provide computer software that enables the radar landmass simulator being developed at NTDC to incorporate data on natural terrain, in addition to the real-time display of cultural targets that was provided for in Contract N61339-69-C-0086. The radar landmass simulator uses TRADEC, the NTDC digital computer system.

The ultimate objective of this effort, of which the current contract is a component, is to advance the technology of using digital and hybrid techniques for radar landmass simulation. The anticipated purpose of a simulator is fleet training in the operation of radar systems and the interpretation of radar displays, in which the flight plan is variable in real time and in which the radar system operator has the full flexibility of the radar panel controls at his disposal. At present the work is oriented toward simulation of conventional scanning radars, although the same techniques can be shown applicable to the higher resolution synthetic aperture (side-looking) airborne radar systems.

The problem area chosen for this simulation is sufficiently large to illustrate and exercise all the techniques of data representation and real-time display reconstruction, yet not so large as to prohibit the development of a simulator with a modest rate of funding. The problem area corresponds to the 1:250,000 scale topographic map sheets Warren, Williamsport, and Scranton, Pa., bounded by latitudes 41°N and 42°N and by longitudes 80°W and 74°W. This area is approximately 60 x 270 nautical miles in extent and encompasses a wide and representative variety of natural terrain and cultural complexes.

The cultural data display in real time previously accomplished is considered Phase 2 in the four-phase development of the radar landmass simulator. The current effort is part of Phase 3 -- the augmentation of the display with

terrain information. This portion, designated herein as Phase 3A, uses general-purpose computer software to simulate the real-time processing that will be performed by the special-purpose computational equipment.

Phase 3A is a computer simulation of the real-time radar landmass simulator; Phase 3B is contemplated to be the embodiment of the unqualified real-time simulation. The Phase 3A display will be produced in real time for view by the observer. The concession is that the computations leading up to preparation of the data for the display will be performed in advance. Thus in Phase 3A the flight plan of the radar being simulated is known prior to writing of the cathode-ray tube indicator, while a simulator designed for fleet training purposes must allow the flight plan to be changed under operator control.

This pre-calculation of information for the display is nevertheless oriented toward the fleet training application: many of the computational approaches and computer programs incorporated in the Phase 3A system will be directly usable in a real-time radar simulator. The purpose of the present development of the radar landmass simulator is to evaluate approaches, develop techniques, and establish guidelines for the eventual procurement of an efficient and effective training device. Hence flexibility is provided in the system being prepared under Phase 3A, and options are incorporated which enable a wide latitude for laboratory experimentation.

Section II gives a statement of the problems and delineates the processing required of each of the programs prepared under this contract. Section III describes the programs that have been written, with a subsection for each of the programming tasks. This descriptive information provides the program structure, input/output formats, arithmetic and logical functions, and operating instructions. Section IV reports on operating experience with the programs (e.g., time and storage requirements, examples of final display, etc.) to the extent this information is available at the time of writing. The detailed documentation of, and microflowcharts for, each of the individual routines and subroutines is given in a series of correlative documents, for each task. These documents are listed in the references (Section V).

SECTION II

STATEMENT OF PROBLEMS

There are five steps to produce the simulated radar pictures of Phase 3A: Reformat Terrain Data, Compress Terrain Data, Produce Sweep Profiles, Produce Intensity Waveform, and Generate Display. The first two prepare the data for the terrain profile generation. This resulting data can also be used in the actual radar simulator. The other three steps simulate the radar simulator. The work was divided into five tasks, corresponding to these five steps in the processing.

1. REFORMAT TERRAIN DATA

To produce a radar landmass simulation system of any significance requires the compilation of a data base for a resonably large area. Radar landmass simulation systems for use in actual training devices will require a data base covering at least half of the continental United States (CONUS). For this effort a minimal area -- that defined by three 1:250,000 scale USGS maps -- of 6 degrees of longitude and 1 degree of latitude was chosen.

Culture data has already been assembled for the chosen area. It is necessary to assemble terrain data from available sources and compile it into a unified data base for use in this study.

The available data on terrain height represents digitization of contours on many separate maps at a scale of 1:250,000. Each map is roughly aligned with the cartesian coordinate system of the digitizer, and the contours are traced by a skilled hand. The digitizer is equipped with a grid of mesh 0.01 inch in both coordinates, and every crossing of a grid line is recorded.

Subsequent computer processing formats a magnetic tape which contains terrain height at every intersection of these 0.01 inch grid lines, as indicated by Figure 1. Those points on the tape that were obtained directly from the contours (closed squares) are tagged differently from those points that have been interpolated (open circles).

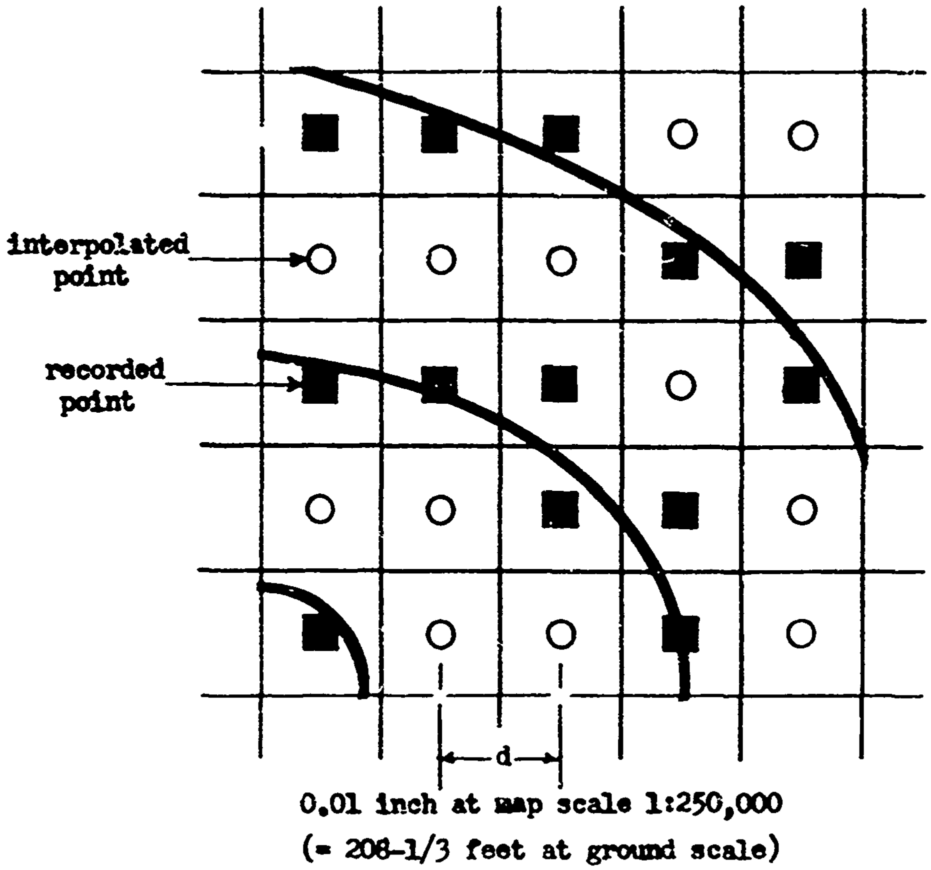


Figure 1 -- CONTOURS AS DIGITIZED FROM MAP (MAGNIFIED 67X)

The U. S. Army Topographic Command (TOPOCOM) is conducting a terrain-digitizing program. This is an on-going program, under the cognizance of the Defense Intelligence Agency (DIA), which is making high-resolution topographic data available in a standard digital format. The source for this digitizing is the 1:250,000 topographic map, the largest scale map that covers the CONUS. The contour intervals for this map series are 20, 50, 100, 200, and 500 feet, depending on the nature of the terrain. All contours are digitized, except those in areas where the steepness of the slope is so great that individual contours cannot be adequately separated. The absolute resolution of the device which converts line (graphic) information to digital form is 0.01 inch, which corresponds to a relative resolution of about 200 feet on the ground. Once digitized and verified, this data is processed by a computer program which interpolates between contours and arranges the information into individual elevations at a density of 0.01 inch over the entire map sheet.

Approximately half of the CONUS is currently in digital form. Included in this area is that portion of CONUS represented by the three 1:250,000 maps WARREN, WILLIAMSPORT, and SCRANTON. These same three maps were digitized by PRA to extract the cultural and hydrograhic information under contract N61339-C-68-0155 (see Ref. 1). After digitizing, the maps were paneled. (Paneling is the term used in the cartographic industry to describe the smooth joining of two adjacent charts in a common coordinate system to form a single chart.)

As supplied by TOPOCOM the digital terrain data for each 1:250,000 map is recorded on two reels of magnetic computer-readable tape: the west half on one tape and the east half on the other. Each half map is in its own coordinate system with its south-west corner near the origin. For west half maps, the east edge is parallel to the y-axis; for east half maps, the west edge is parallel to the y-axis. Each half map is approximately a trapezoid (the north and south edges are slightly curved). Figure 2 shows an exaggerated diagram of two half maps, with their coordinate systems, and gives typical dimensions in inches.

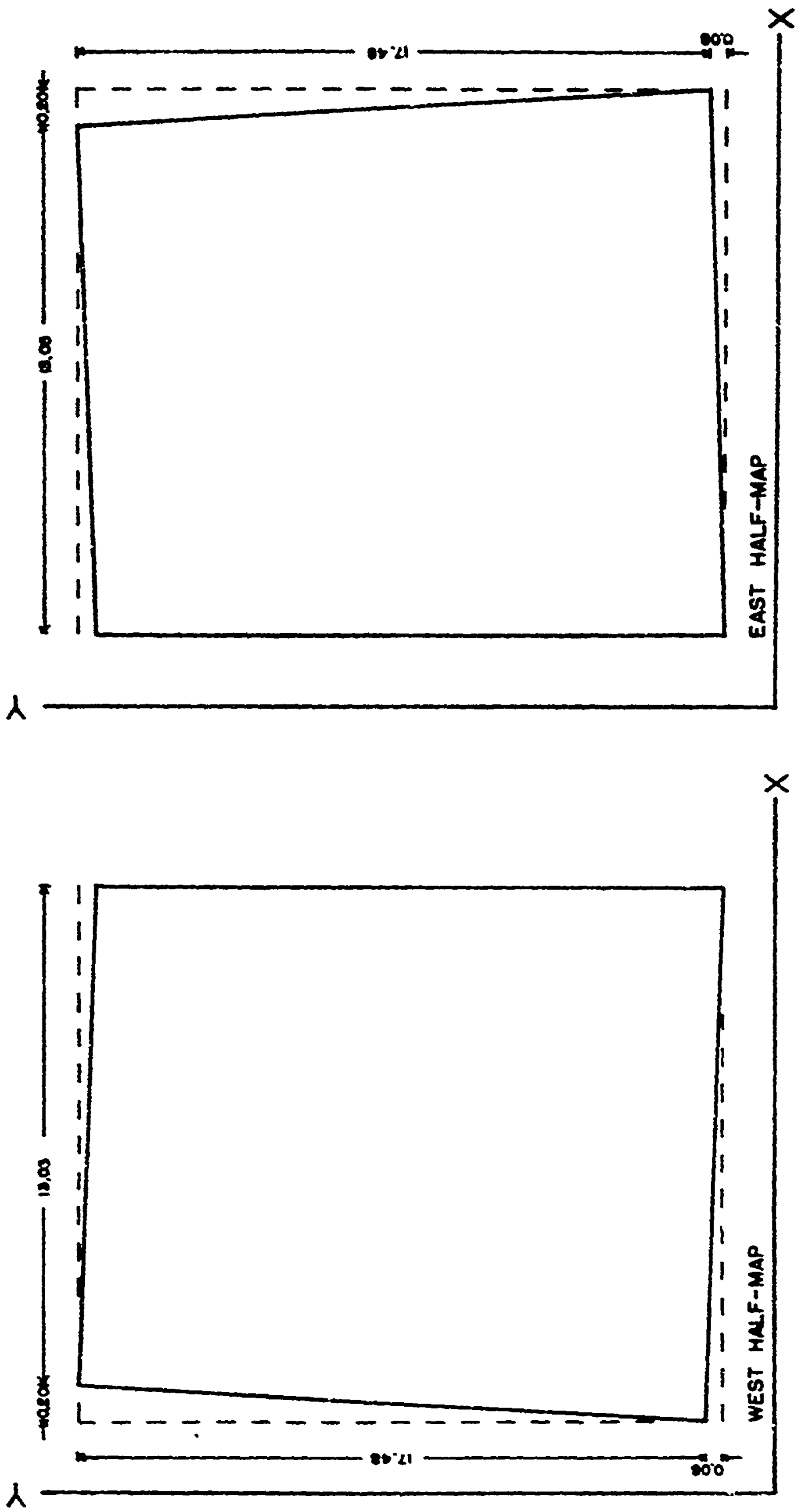


Figure 2 -- TWO HALF-MAPS WITH THEIR COORDINATE SYSTEMS
(dimensions in inches)

The x- and y-coordinates are represented by 12 bits (0.01 inch out of 40.96 inches). The height, or z-coordinate, is represented by a sign and 15 bits (1 foot out of ± 32,768 feet) along with two bits which distinguish recorded contour points and map-edge points from computer-interpolated points.

The grid spacing of 0.01 inch corresponds to 2,500 inches or 208-1/3 feet at ground scale: this is the sample point spacing, in the coordinate system in which the map was digitized. When related to the unified coordinate system, the three maps of Figure 3 form a data base contained within a rectangle 60 x 270 (nautical) miles.

The coordinate system chosen for the culture data has its origin at the south-west corner of the WARREN (west most) map. The y-axis is along the west edge of the WARREN map (i.e., along the 80° longitude line). The x-axis is perpendicular to the y-axis and is tangent to the south edge of the WARREN map at its south-west corner. Figure 3 gives an exaggerated diagram of the paneled six half-maps in this coordinate system and gives dimensions in inches. Section III.1.2 describes the transformations.

The data is organized on the tape as 18-bit words. This organization is not convenient for processing on a computer with a 32-bit word. The job of Task 1 is to panel the six half-maps into a unified data base using the same coordinate system used for the culture data, and incidental to this paneling, Task 1 should reorganize the data into a format more convenient for processing on a computer with a 32-bit word.

2. COMPRESS TERRAIN DATA

This section summarizes the considerations entering into the representation of natural terrain for the purposes of real-time digital simulation of radar displays. It serves also as a primer on approaches for finding a method by which natural terrain can be represented in a computer-readable form that is efficient for use by a real-time digital simulator.

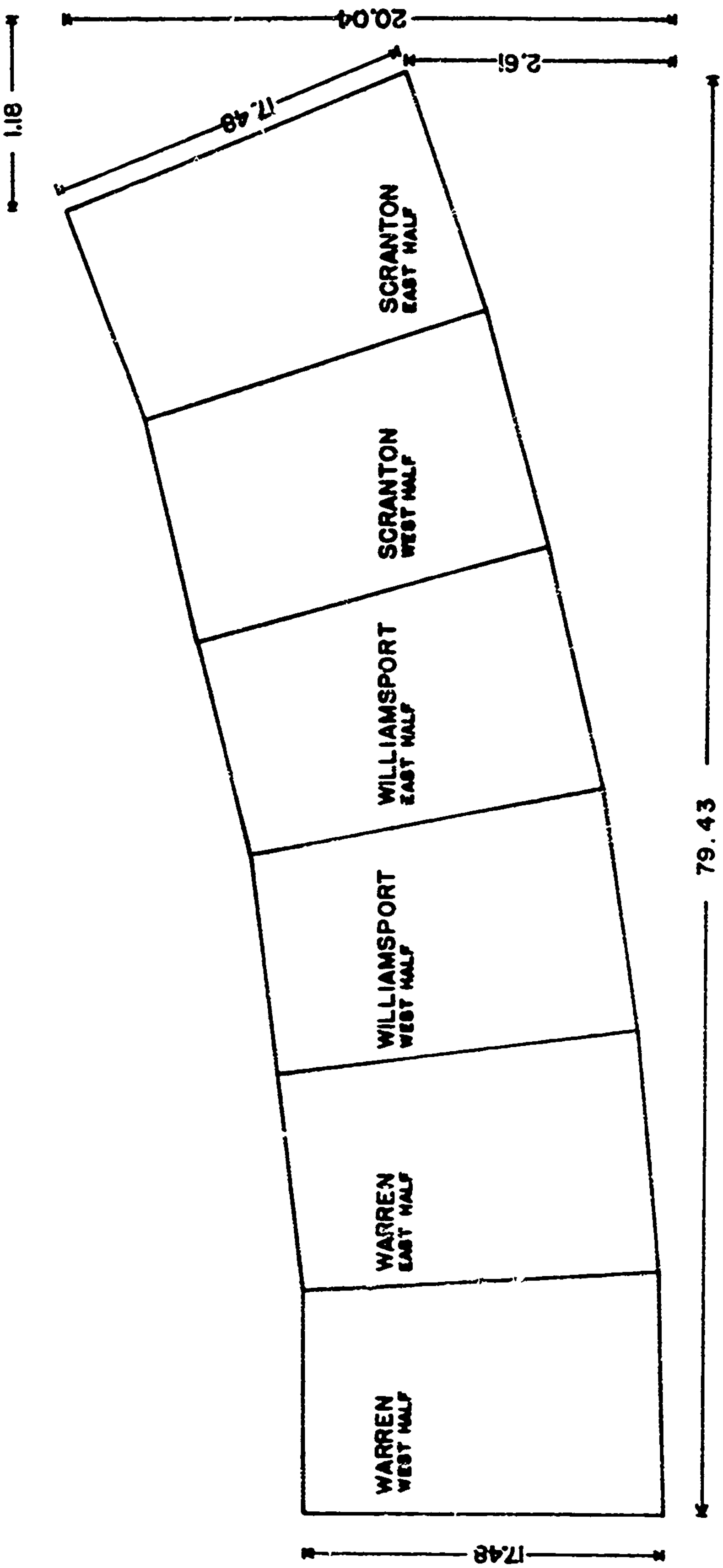


Figure 3 -- THE SIX HALF-MAPS IN THE UNIFIED COORDINATE SYSTEM (EXAGGERATED)
(dimensions in inches)

The "radar landmass simulator" provides a display that appears the same as the display the observer would see if he were operating a radar set in an actual aircraft. The simulator design team is given a <u>data base</u> which describes the terrain in cartographic terms (generally contour maps). The real-time system is provided with a <u>simulator map</u> which is derived from the information in the data base and is stored in a suitably encoded and compressed form.

Preparation of the simulator map is strongly dependent on the processing that will be employed in the simulator to reconstruct terrain height profiles along each radar sweep within the scan. For the present project, the scan format is the plan position indicator (PPI), and radar sweeps are equispaced radial lines originating from the current position of the vehicle in which the radar is mounted. Computational techniques suitable for terrain reconstruction have been studied at some length in previous projects. Straightforward tabulation of terrain height and various methods of compressing such a tabulation have been investigated in the past. It is currently believed that a functional approximation will allow the simulator to take best advantage of the redundancy (smoothness, regularity, consistency, etc.) inherent in the terrain height as a function of lateral position.

Several serious considerations underlie the terrain fitting problem. Among these are the computer-oriented limitations on storage size, processing speed, and transfer rate, which are directly translatable into dollars per unit area of real estate (for data preparation) and dollars per simulator (for fabrication of the device proper). Further, as a pictorial display the simulated radar scope must cater to the capabilities and vagaries of the human observer: disturbing patterns or unrealistic effects cannot be allowed, and the presented image must look radar-like regardless of any convenient mathematical formulation that may be found. These factors reduce the number of variables or options in the terrain fitting problem and, in turn, give rise to some constraints that must be taken into account in making the (digital) simulator map.

2.1 COMPUTER HANDLING OF SIMULATOR MAP. The computer processing to reconstruct terrain height is going to be repeated for every resolvable element along a radar sweep line, which line may have its end points located anywhere within the problem area. It is intuitively obvious that the burden on the computer will be alleviated if one or more data groupings are made, intermediate between the resolvable-element level and the problem area level. The area of terrain that corresponds to a group of resolvable elements, having a specific size and shape, is termed a region. For simplicity, both the area covered by a region and the data that characterizes that area are frequently referred to as regions. The simulator map, to be prepared from the data base described above, should group the data by regions for ease of computer handling.

Further developing the concept, one recognizes the convenience of computer handling that arises from making a region's dimension a binary multiple (2,4,8,16,...) of the dimension of a resolvable element. In a digital computer the digits defining the coordinates of a region are the high order digits, and those defining the location of a resolvable element within the region are the low order digits, with no overlap. This makes addressing calculations simpler both to program and to execute. For example, as a radar sweep is traced out by counting resolvable elements from some starting value (x_{min}, y_{min}) to some ending value (x_{max}, y_{max}), the high order digits of the x-counter and y-counter determine the region in which the sweep lies at any instant. Further, the repetitive nature of the low order digits in the counters is useful in generating signals which depend only on sweep location within a region.

This method is diagrammed in Figure 4 which illustrates the case of 16 resolvable elements in each axis, grouped into four regions in each axis. Two x-bits and two y-bits determine the region, and two more x-bits and two more y-bits determine the resolvable element within the region. The heavy outline on Figure 4 bounds region (x = 10--, y = 01--), and the resolvable element (hereafter called a dot) illustrated takes position (x = --10, y = --11) within that region. Hence, the complete coordinates of that dot are (1010,0111).

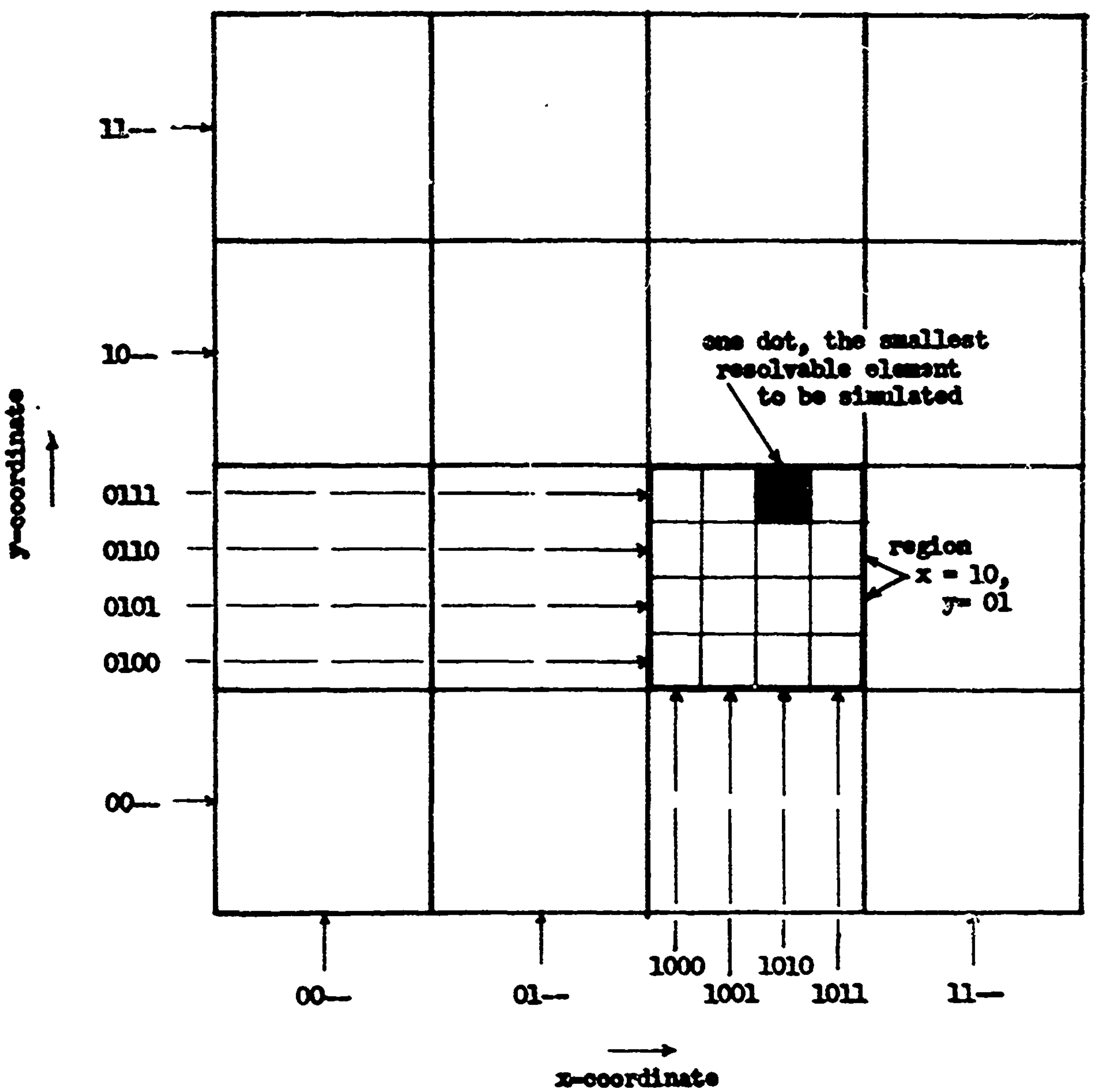


Figure 4 -- THE PROBLEM AREA AS DOTS WITHIN REGIONS

In previous work, the standard has been adopted that a region contains 32 x 32 dots, or five bits for the low order part of the x- and y-coordinates. Further, the region dimension has been set at 5,000 feet, corresponding to a dot size (resolution capacity) of 5,000/32 = 156.25 feet. The data on culture for the same problem area has previously been encoded in a unified coordinate system using increments of 156.25 feet, and computer programs to handle culture data in this form are in existence.

The region dimension is thus exactly 24 times the grid line spacing, although the grid lines of any digitized map are skewed with respect to the final simulator map. Regions on a 1:250,000 scale map can be visualized by looking through a quarter inch aperture. (To be exact a 0.24 x 0.24 inch grid is necessary.)

A region must comprise an independent packet of data, so the terrain-reconstructing computer may handle it individually. The choice of 5,000-foot regions is a compromise between small regions that would require an excessive number of storage accesses, and large regions that require excessive amounts of data to be handled per storage access. This dimension corresponds to just over 10 microseconds of two-way signal travel across the terrain.

The mathematical notation reflects the grouping of data by regions. Capital letters are used to denote regions, and small letters are used to denote position within a region. Thus, the dot coordinates (X+x, Y+y) imply region (X,Y). As mentioned above, five bits is sufficient to represent x or y. For the problem area presently used, X requires seven bits. (It may be of interest to note that 12 bits for X and 5 bits for x allows representation of 4,096 regions, or 20,480 feet, or about 3,350 (nautical) miles -- more than adequate to span the continental U.S.A.) The radix point is considered to be placed so that $0 \leq \binom{x}{y} < 1$, and the region coordinates X,Y are integers.

2.2 PROBLEMS OF ARTIFICIAL PATTERNS. The computer processing may tend to introduce artificial patterns, as from roundoff or truncation error at regular intervals. Even if such errors are small compared to the error in the fitting process, their regularity will provide visual cues that -- in the final reconstructed image -- will be disturbing and thus unrealistic.

Such cues may take the form of moire patterns, resulting from the beating of two, individually invisible, sets of quantization steps . A well-known example from previous work is the moire effect caused by calculations made in a polar coordinate system being displayed on a cartesian raster of comparable resolution.

Granularity in the final display will arise if insufficient precision is used for the intensity coordinate. The eye will be drawn to patches of relatively constant intensity with well-defined boundaries. For smoothly varying radar intensity, these granularity boundaries take the form of contours: the terrain will be terraced. To avoid contouring, the intensity coordinate is frequently made an order of magnitude more precise than its accuracy or error size would require. That is, several low order bits must be carried for purposes of smoothness, even if there is no justification for that number of significant figures in terms of accuracy of fit.

Checkerboard patterns are particularly likely to result from the grouping of resolvable elements into regions described above. If small errors are made in the height of each region, there will be bluffs or escarpments at the region boundaries which, of course, will show up as bright lines or shadows in a most regular fashion. Moreover, the radar display is sensitive to slope. Slope discontinuities in the terrain are transformed into intensity discontinuities in the display. These would appear as contours of intensity, here aligned nicely with the region boundaries to give the impression of a grid superposed over the radar display.

A mathematical formulation of boundary continuity is appropriate. Let the actual terrain height z be approximated by the function $h_{X,Y}(x,y)$. From the previous discussion of Figure 4, this means the approximated height h in region (X,Y), at position (x,y). The criterion of boundary continuity in height and slope is then expressed as:

$$\frac{\partial^s}{\partial x^s} h_{X,Y}(0,y) = \frac{\partial^s}{\partial x^s} h_{X-1,Y}(1,y) \quad \text{for all} \quad y = (0,1)$$

$$\frac{\partial^s}{\partial y^s} h_{X,Y}(x,0) = \frac{\partial^s}{\partial y^s} h_{X,Y-1}(x,1) \quad \text{for all} \quad x = (0,1)$$

where $s = 0,1$ (height and gradient).

This is illustrated in Figure 5, where the arrows show possible radar sweeps across region boundaries. The criterion of boundary continuity must be imposed, whatever the fitting scheme may be, to avoid the checkerboard pattern (in addition to the above criteria relating to moire patterns and granularity). One way of doing so is illustrated by Figure 6. Here four sets of alternating nonadjacent regions are individually fitted and the fits are combined. If the fit to each region includes some area in the adjacent regions, then the average fit will be reasonable and will exhibit suitable continuity.

2.3 SIMPLE RECONSTRUCTION. It is desired that the terrain height reconstruction be computationally simple. While "simplicity" is an intuitive rating, certain basic properties of the reconstruction process will be readily agreed as leading toward a simpler mechanization.

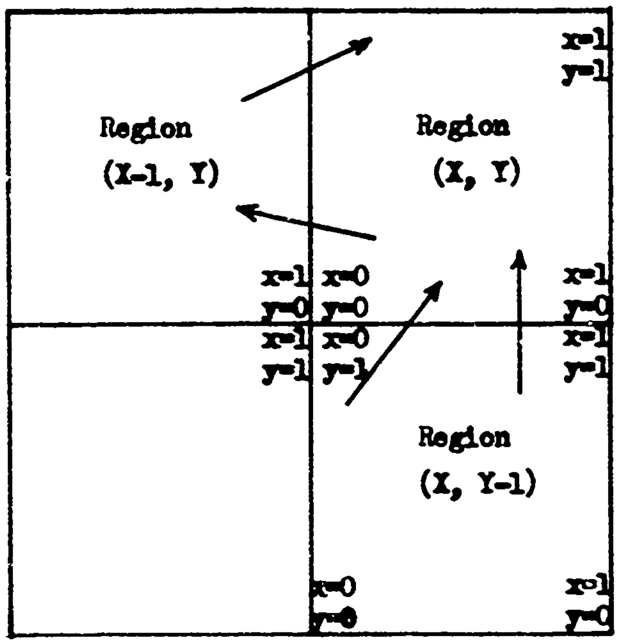
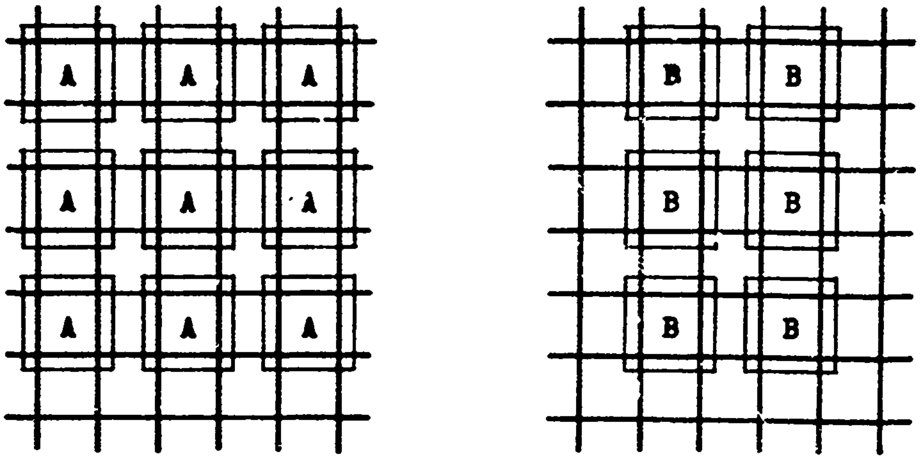


Figure 5 -- ILLUSTRATION OF CONTINUITY REQUIREMENTS



.... and so on for the 'C' and the 'D' sets of regions.

For each set of regions, the coefficients of the simulator map are found by taking the area within and nearby the regions comprising the set. This provides four sets of coefficients, which are assigned to region corners. The final coefficients are found by (weighted?) averaging of the four sets of coefficients, at corresponding corners. In other words, a four-color checkerboard is conceived. Each color (A, B, C, D) is treated separately, and coefficients for the respective regions are computed, each region being a bit larger than originally assigned (overlapping the regions of the other colors, as shown). Then coefficients for each corner of the original size regions are found by averaging the respective coeficients computed for the four oversize regions which include each corner.

Figure 6 -- FITTING TO ALTERNATING SETS OF REGIONS

The first such property is superposition. If the approximated terrain height at any point can be found by the superposition of independently calculated functions, such calculations may proceed in parallel (perhaps by analog computing gear) and summed for the resultant value of height. That is, a reconstruction formula in the following form is preferred:

$$h(x,y) = \sum_i c_i \, g_i(x,y)$$

where the set of coefficients c_i is appropriate to the region (X,Y) being reconstructed. The approximation is thus a linear combination of the basic functions g_i.

The second property is separability. Since the terrain is arbitrarily located with respect to the problem area coordinate system, symmetry should exist between the x- and y-coordinates. Separability implies that the independent calculation of functions that are superposed, be extended to independence between these coordinates. Thus the following form is assumed to be "simpler" than the above:

$$h(x,y) = \sum_i \sum_j [c_i \, g_i(x)] \; [c_j \, g_j(y)]$$

$$= \sum_i \sum_j c_{ij} \, g_i(x) \, g_j(y)$$

where the set of coefficients c_{ij} is appropriate to the region (X,Y), and $g_i = g_j$ for $i = j$, with any argument x or y.

The third property leading toward simple methods of reconstruction arises from a recognition that the criterion of boundary continuity places some constraints on the coefficients in adjacent regions. Hence coefficients for adjacent regions are somehow related and could be combined to serve

more than one region. The desired property is thus coefficient sharing, or the use of coefficients by more than one region, leading to less total storage and less data transfer.

Carrying this property further, one may conceive of coefficients being assigned to corners of regions, to be shared by the four regions adjacent to each corner. There are as many corners as regions in the problem area;* so this scheme provides as many sets of coefficients as regions. However, each region is reconstructed from four sets of coefficients, and passing across a region boundary requires changing only two of these sets of coefficients.

The fourth property which promotes a simple means of reconstructing the approximated height is the ready computation of each of the $g_i(x)$ and $g_j(y)$. That is, the form of the basis functions should be easily amenable to numerical evaluation on a computer. As an example, polynomial forms have been used: $g(x) = a_0 + a_1x + a_2x^2 + \ldots$, and the various specialized forms of polynomials (Legendre, Tchebyschev, Lagrange, etc.) have differing characteristics that are worthy of consideration.

Since the basis functions are the same for all regions, their calculation may be replaced by table lookup. To this extent, simplicity of the basis functions themselves is not so important. If the region size is 32 dots on a side, as previously standardized, and if the basis functions are separable into functions of x and functions of y, then each basis function requires at most 32 table entries. (Lack of separability would mean that each basis function $g_i(x,y)$ might have $32 \times 32 = 1024$ table entries, although a smaller number of different functions g_i, would suffice.) Tabulation of Lagrange polynomials has been effective in past work.

---

*Each region may be considered to include its left and bottom boundary but not its right or top boundary; this is equivalent to defining a region as the area within the interval $x = (0,1]$, $y = (0,1]$. Then regions may be named by the coordinates (X,Y) of their lower left corners.

2.4 THE FITTING PROBLEM. In view of the foregoing considerations, the problem of terrain fitting may be stated as: Given the actual height z as described in Section II.1, find a method of fitting z which allows reconstruction of an approximate height h on a region by region basis as described in Section II.2.1. There must be continuity of height and slope across the region boundaries as described in Section II.2.2, and the several desirable properties described in Section II.2.3 should apply. The approximation can be expressed as:

$$z \approx h_{X,Y}(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{n} c_{X,Y,i,j}\, g_i(x;\lambda)\, g_j(y;\lambda)$$

where X,Y = regions designator (integer part of lateral coordinates)
x,y = position in region (fractional part of lateral coordinates)
i,j = indices of summation, corresponding to order of g_i, g_j
$c_{X,Y,i,j}$ = coefficients of order i,j (applying to region X,Y)
$g_i(x;\lambda)$, $g_j(y;\lambda)$ = basis functions of orders i,j (common to all regions)
n = degree of approximation (indicative of number of coeffients used)
λ = specific set of basis function (e.g., Lagrange polynomials.)

The form of the basis functions should be selected to (a) minimize the amount of data contained in storage and transferred during reconstruction and (b) maximize the accuracy of fitting relative to some appropriate criterion. Obviously this represents a tradeoff, and this tradeoff is the fundamental problem of terrain fitting. Data minimization (or data compression is affected by:

- the degree of approximation n
- the number of bits carried in the coefficients c
- the extent to which coefficients apply to more than one region.

The problem of data compression is easily recognized.

The accuracy, or fidelity, criterion is more complicated than has been recognized in previous work. Although reduction of rms error between z and h is mathematically convenient, it is believed that a re-examination is needed of the relation between some height metric and what the radar observer sees.

2.5 RADAR-ORIENTED FITTING CRITERION. Regardless of how well the fitting is done, the appearance on the observer's PPI is all that matters. Discussion of artificial patterns above suggested that the observer's eye is sensitive to intensity changes, with absolute intensity levels of little significance. In light of the variation allowed in the radar set's contrast and sensitivity controls, it would be unnecessarily confining to attempt an accurate match of terrain height. Instead, the <u>changes</u> in scope intensity should be matched -- which correspond more or less to the second derivative of terrain height as a function of lateral coordinates. (Intensity corresponds to terrain slope, except for shadows and other objects where they occur.)

The airborne radar observer is trained to identify locations of ridge and drainage lines in the terrain and to interpret topographic features by reference to their intensity contours and shadow boundaries. He looks for cues such as angularity vs. smoothness in the lateral sense, and such as sharpness vs. diffuseness in intensity. He attaches little significance to variations in intensity level that are small compared to the average variation within the area of scrutiny. The radar is usually viewing the terrain at low elevation angle, and the pattern is changing with motion of the vehicle in which the radar is located. Sharp convexities, cliffs, and scarps are prominent regardless of radar position or panel control settings, while flat valleys, plateaus, and low rolling hills tend to be masked by noise or by hydrographic and cultural features.

These properties must be used in the formulation of a fitting criterion. The criterion must take into account the different kinds of terrain: a 200-foot hill in the flat midwest is significant, whereas it is irrelevant

in the western mountains. Consideration has been given to a criterion that is related to spatial frequency content of the terrain fluctuation (where the high-amplitude components are weighted more heavily), and to a criterion that places emphasis on the second derivative of terrain height. Further work is required to formulate more specific fitting criteria having radar significance, and to relate subjective pictorial accuracy to a mathematically expressed fitting criterion.

2.6 **PROGRAM TO BE WRITTEN.** The job of Task 2 is to prepare a reasonably generalized terrain compression program which compresses the terrain into a set of coefficients for each region as described above. Variation is allowed in the following parameters:

- Basis Function. A subroutine is provided to tabulate the basis functions of a single variable. Only the separable form is provided for. Up to ten basis functions may be used.

- Region Size. Any integral multiple of the grid lines spacing of the original data may be used.

- Degree of Overlap. The concept of a group which surrounds a region is introduced. The region is centered within its associated group, and the group size may be anywhere from the region size to twice the region size.

- Problem Area Size and Location. Any rectangular problem area may be defined. The units are grid spacings within the original data. The problem area may be wholly within the area covered by the data, the data may be wholly within the problem area, or there may be some of the problem area not containing data and data not contained within the problem area. Coefficients are computed only for those regions whose associated group is wholly contained within the area covered by the data. All other regions have zeros for their coefficients.

- Coefficient Precision. A maximum of 15 significant bits plus sign is output from this program. The power of two represented by the most significant bit and by the least significant bit may take any values provided that no more than 15 significant bits lie between the most significant and least significant.

- Coefficient Sharing. If the degree of representation is denoted by N, there are N^2 coefficients for each region. One quarter of these N^2 coefficients is associated with each corner. To provide for boundary continuity as described above in addition to the criterion placed upon the functions, the corresponding coefficients must be equal between neighboring regions. For example, the coefficient which is multiplied by those two functions which take on the value unit at the lower right corner of a region must equal that coefficient which is multiplied by those functions which take on the value unit in the lower left corner of the next region to the right. The program allows for any subset of the $N^2/4$ coefficients to be averaged with the coefficients in neighboring regions and the resulting average used for both regions. Therefore, any degree of boundary continuity may be incorporated within the resulting simulator map.

Section III.2.3.2 gives details on how the above parameters are specified. The limits mentioned above indicate the maximum for each particular parameter. However, the exercising of all options at their maximum value is not possible. Associated with each value of the above parameters is a core storage requirement. The set of parameters chosen must be such that the total available core memory is not exceeded. Section III.2.3.3 tells how to compute the required storage for any given set of parameters.

3. PRODUCE SWEEP PROFILES

The radar landmass simulator designed by PRA under previous contracts (see Refs. 2, 3, and 4) can be diagrammed by Figure 7. It consists of three sections: the culture profile generator, the terrain profile generator, and the radar effects generator. These three are controlled by and receive their data from the general-purpose digital computer. As described in Ref. 2 the computer software is organized in a multiprogramming structure. There is a special-purpose control program, five primary tasks*, and secondary tasks as needed. The secondary tasks interface the input/output units with the primary tasks; the primary tasks perform the actual control function of the simulator system. The names and functions of the five primary tasks are as follows:

- MISCON simulates aircraft motion,
- TFCONT produces height profiles for a scan,
- CFCONT produces reflectance profiles for a scan,
- TFPREP updates the terrain data window
- CFPREP updates the culture data window.

Figure 8 shows the system cycle. While the height profiles are written, the culture window is updated as appropriate. Then, while the reflectance profiles are written, the terrain window is updated and the aircraft position is updated. In addition to updating the culture data window, CFPREP also tries to keep ahead of the aircraft by predicting which strip of culture and terrain data will be needed for the next update. Prediction is possible because the window is sufficiently larger than the scan. Updating is then done either when the data is available (usually one scan after it was requested), or when the current window cannot hold the current scan. The terrain window is updated whenever the culture window is updated.

*Both the RLMS system software and the project management are divided into "tasks." An RLMS system software task is one or more programs which perform some function within the simulator. A project management task, on the other hand, is one or more jobs to be performed as specified in the project work plan.

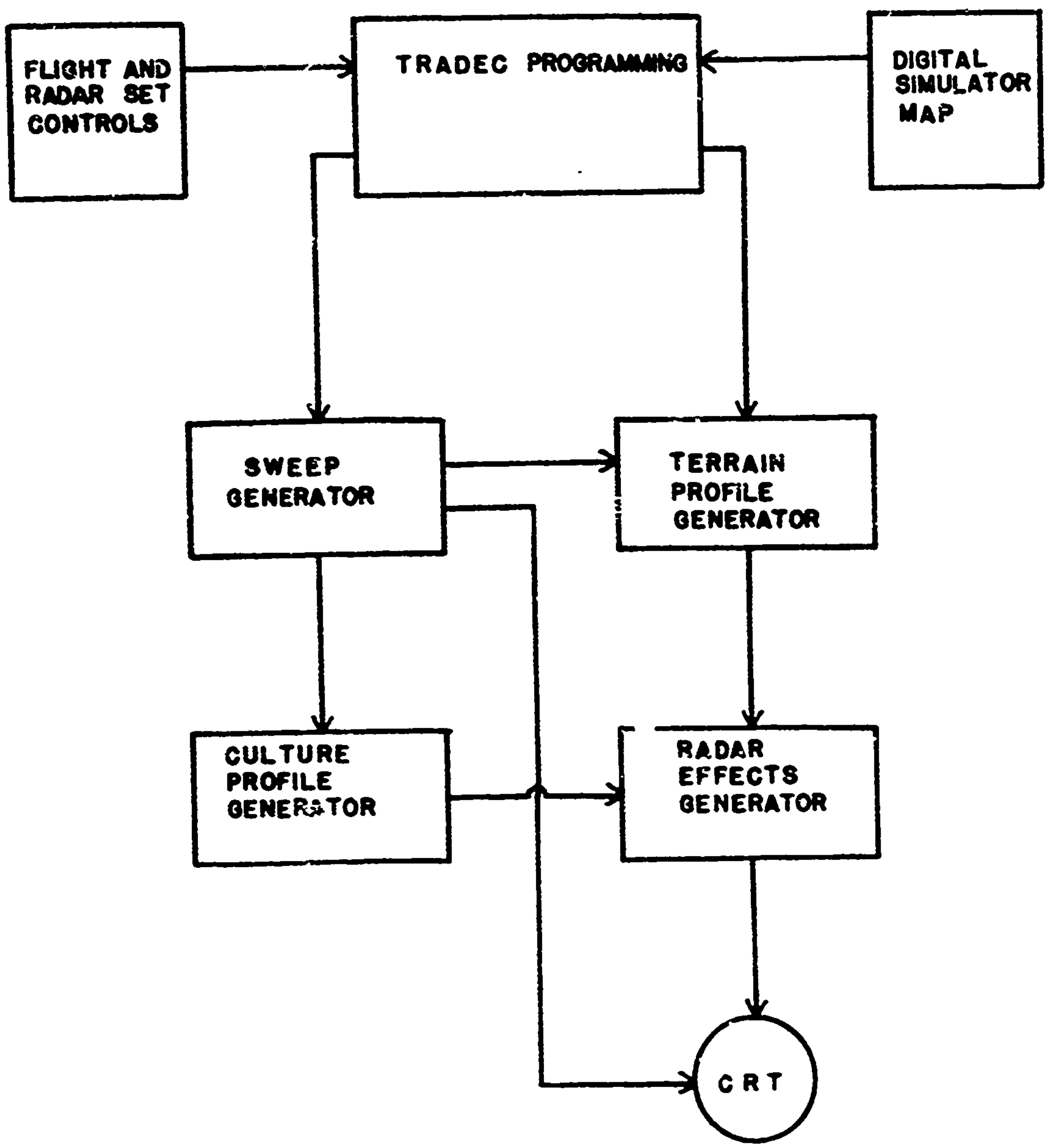


Figure 7 -- DIAGRAM OF RLMS SYSTEM

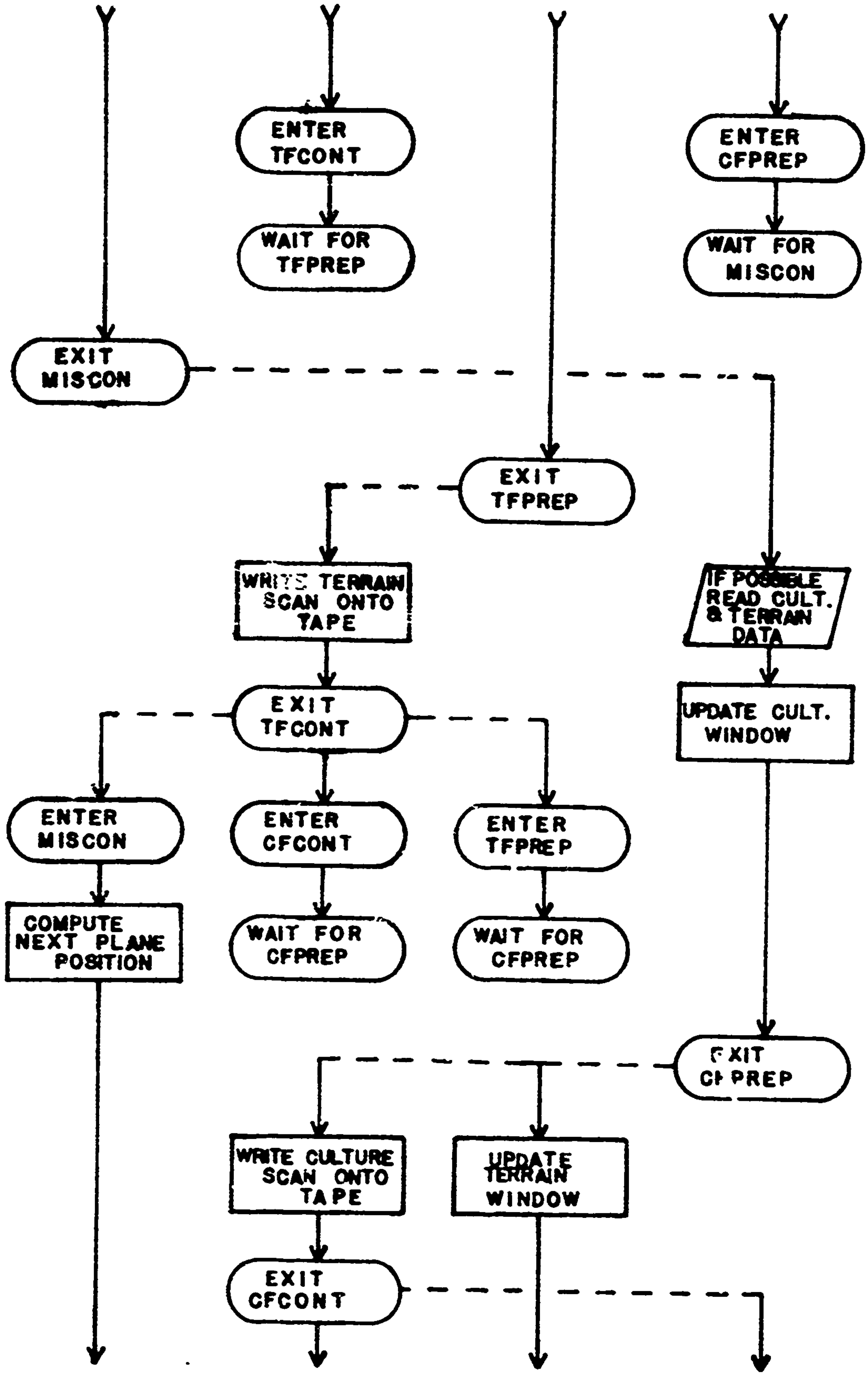


Figure 8 -- PHASE 3. SYSTEM CYCLE

Figure 9 shows the initial system cycle. The system begins by starting MISCON, CFCONT, TFPREP, and CFCONT. CFCONT and TFPREP wait for CFPREP; CFPREP waits for MISCON. MISCON calculates the initial plane position. CFPREP then creates the initial culture window. CFCONT and TFPREP in parallel write the culture scan onto tape and create the initial terrain window. When CFCONT exits, TFCONT and CFPREP are entered.

The system is now ready to operate as shown in Figure 8. CFCONT and CFPREP begin operating in parallel. Before CFPREP can proceed it must wait for MISCON to finish. Before TFCONT can proceed, it must wait for TFPREP to finish. Both MISCON and TFPREP were entered during the previous cycle. The waiting is included in the programs as a safety factor.

TFCONT and CFPREP is parallel write the terrain scan onto tape and update the culture window. When TFCONT exits, MISCON, CFCONT, and TFPREP are entered. CFCONT and TFPREP must wait for CFPREP to finish. Again, this waiting is included as a safety factor. MISCON, CFCONT, and TFPREP will, in parallel update the plane position, write the culture scan onto tape, and update the terrain window. When CFCONT exits, TFCONT and CFPREP are again entered, starting the cycle over again.

A radar landmass simulation system which consists of the culture profile generator, display, and general-purpose computer is currently operational at NTDC. Under Contract N61339-69-C-0086 PRA designed and implemented the software for this system consisting of:

- The special-purpose multiprogramming control program.
- All necessary secondary tasks.
- The primary tasks MISCON and CFPREP.

This system is described in detail in Ref. 4. As stated in Section I this is known as the Phase 2 system. The Phase 3 system will include the hardware and software to produce terrain profiles and combine the two profiles into

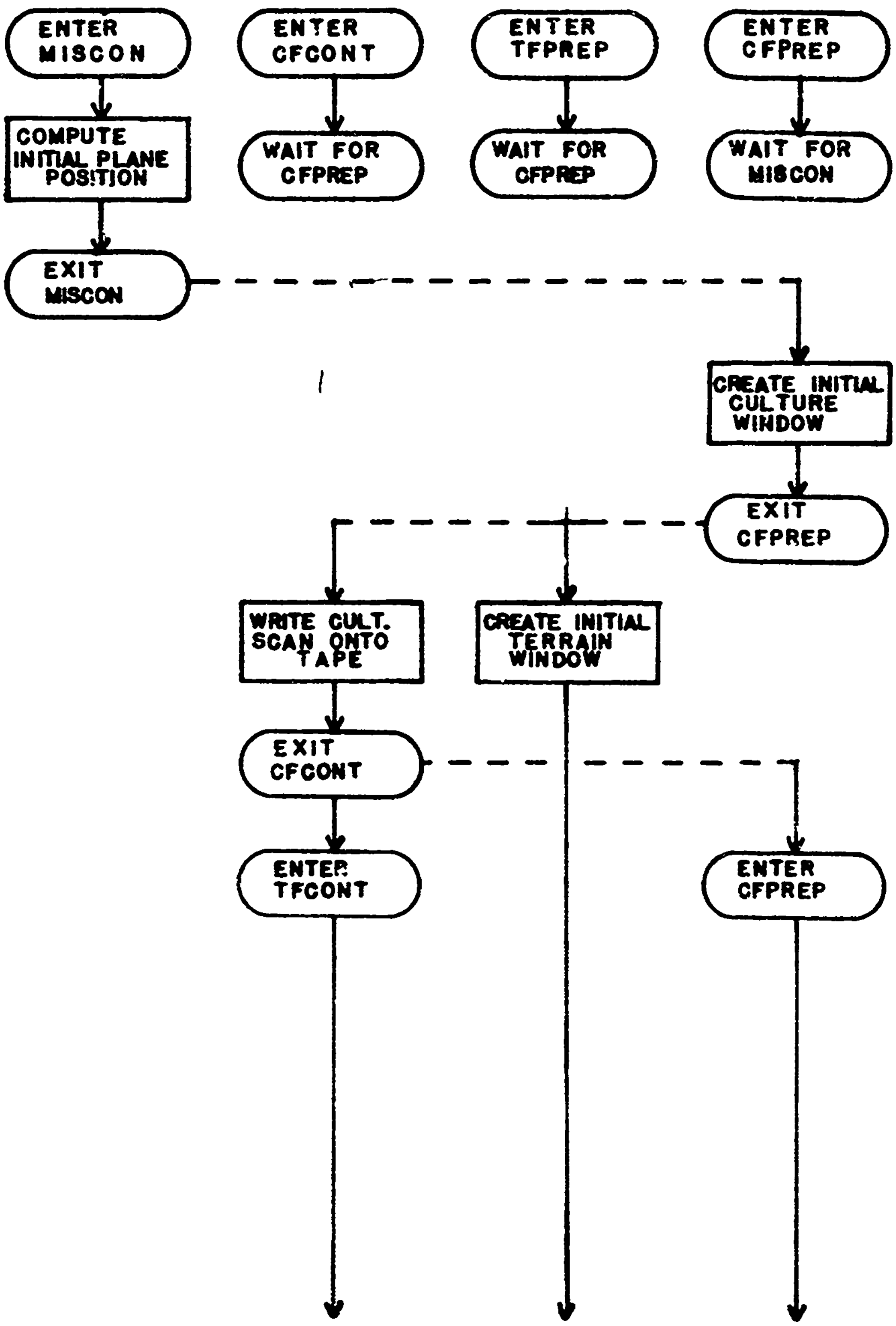


Figure 9 -- PHASE 3A INITIAL SYSTEM CYCLE

a radar picture. Numerous questions were raised in connection with the detailed design of the Phase 3 system (see Ref. 6); therefore Phase 3A was included in the program to simulate the proposed special-purpose hardware, prior to performing Phase 3B which will satisfy the original Phase 3 concept.

The purpose of Task 3 is to develop those computer programs which calculate terrain height and radar reflectance profiles as a function of ground range along each of the sweeps comprising a radar scan. Those portions of the Task 3 programs that have to do with calling the terrain data from the simulator map form the prototypes of the real-time programs to perform the corresponding operations. Those portions of the Task 3 programs that simulate the special-purpose hardware perform calculations that faithfully represent the design of the special-purpose hardware.

4. PRODUCE INTENSITY WAVEFORM

Previous design work has de-emphasized the radar effects generator portion of the radar landmass simulator. Satisfactory radar effects generators exist within factored transparency simulators. NTDC directed that PRA concentrate its efforts on those aspects -- in particular the terrain profile generator -- of the radar landmass simulation for which unique technological innovations and significant resulting improvements were contemplated over the factored transparency approach.

The job of Task 4 is to write a program which will take the culture and terrain profiles and produce an intensity profile similar to that produced by the radar effects generator. The steps involved in this are as follows:

- Perform smoothing of the terrain height profile to simulate removal of quantization noise. The degree of smoothing will be pre-set by an assembly parameter to this program.

- Mix terrain height profiles with culture reflectance profiles, taking account of Lambert's law, antenna elevation pattern, and terrain shadowing

- Perform ground-to-slant range transformation, including atmospheric refraction and earth curvature

- Compress the resulting slant range intensity profile, representing the signal to be applied to the display, to a form suitable for later access in real time, and store on tape and/or disc

- Perform such control and housekeeping tasks as are necessary to provide for effective operation of the above programs.

Section III.5 describes the programming structure and mathematical techniques to accomplish the above.

5. GENERATE DISPLAY

The purpose of the Task 5 program is simply to read the data from tape and/or disc that has been previously calculated, and to control the special-purpose display hardware constructed by NTDC for driving the cathode-ray tube indicator. This is to be a "load and go" program, suitable for demonstration of radar simulation capabilities with little or no preparation on the part of the computer operator. Thus even though the data is prepared by a non-real-time computation, the Task 5 program provides output to the display equipment in a time frame corresponding to the scanning sequence of a radar, irrespective of the amount of computation in Tasks 3 and 4 that was necessary to support such a display. As mentioned above, only the scan format of a 20-mile range, 45° sector is provided at the present time. Computer programs within Task 5:

- Provide information regarding position and orientation of each sweep for transmission to display hardware

- Read the intensity profile from tape or disc for each sweep and transmit it to the special-purpose hardware for display writing

- Provide necessary control and acknowledgement signals via the computer input/output channel required to interface with the display hardware

- Perform such control and housekeeping tasks as are necessary for effective operation of the above programs.

SECTION III

DESCRIPTION OF PROGRAMS

Section II introduced the five tasks in Phase 3A of the development of a hybrid radar landmass simulator. The job of each of these tasks was to write a program. This section contains the overall description of each of these programs and their user documentation. Detailed programming documentation is provided in separate documents Refs. 8-12.

1. REFORMAT TERRAIN DATA

The job of Task 1 is to accept the six tapes representing the WARREN, WILLIAMSPORT, and SCRANTON maps and create a unified map of the same problem area. The program is specifically tailored to process these six tapes. However, it is possible to process six similar tapes with this program.

This section contains a description of the overall program structure, followed by a description of the program flow. Then the "User's Manual" for this program is presented.

1.1 PROGRAM STRUCTURE. The Task 1 program consists of a main program and nine subroutines. Figure 10 is the call tree for this program. It shows which subroutines call which other subroutines and which subroutines are called by which subroutines. For example, the subroutine with the main entry point PROEI calls the subroutine with the main entry point RDDC and is called by the executive and the subroutines with the main entry points CHECKPNT and PROD.

The executive does none of the actual processing; rather it calls upon six of the nine subroutines in the proper order. These six subroutines do the actual processing with the aid of the other three subroutines. Figure 11 shows these six subroutines and the interfaces among them.

PROA takes the coordinate definitions of corners of each of the six half-maps and computes the transformation parameters.

PROB reads one column of data from the TOPOCOM data tapes each time it is called. This column of heights is unpacked and stored one height per word into an array. The (X,Y) coordinates of this column are also extracted from the data tape and unpacked.

PROC applies the transformation parameters to the X,Y coordinates for the current column to compute the new coordinates (X_1, Y_1) for each height in the current column. These coordinates are stored into two arrays.

PROD uses the (X_1, Y_1) coordinates to store the column of heights into an intermediate array on the disc. Depending upon which half map it came from, a column of data may contribute to as few as 19 columns in the final output or as many as 118 columns in the final output. The intermediate array contains 120 columns.

PROEI empties one column at a time from the intermediate array and reformats it into final output. It also acts as an interface between the disc and the other routines.

CHECKPNT saves the transformation parameters and the intermediate array onto the checkpoint tape. This is to allow stopping and restarting the program at intermediate points. It also allows saving intermediate results for restarting in case of machine failure.

Detailed documentation of all programs constituting Task 1 may be found in Ref. 8.

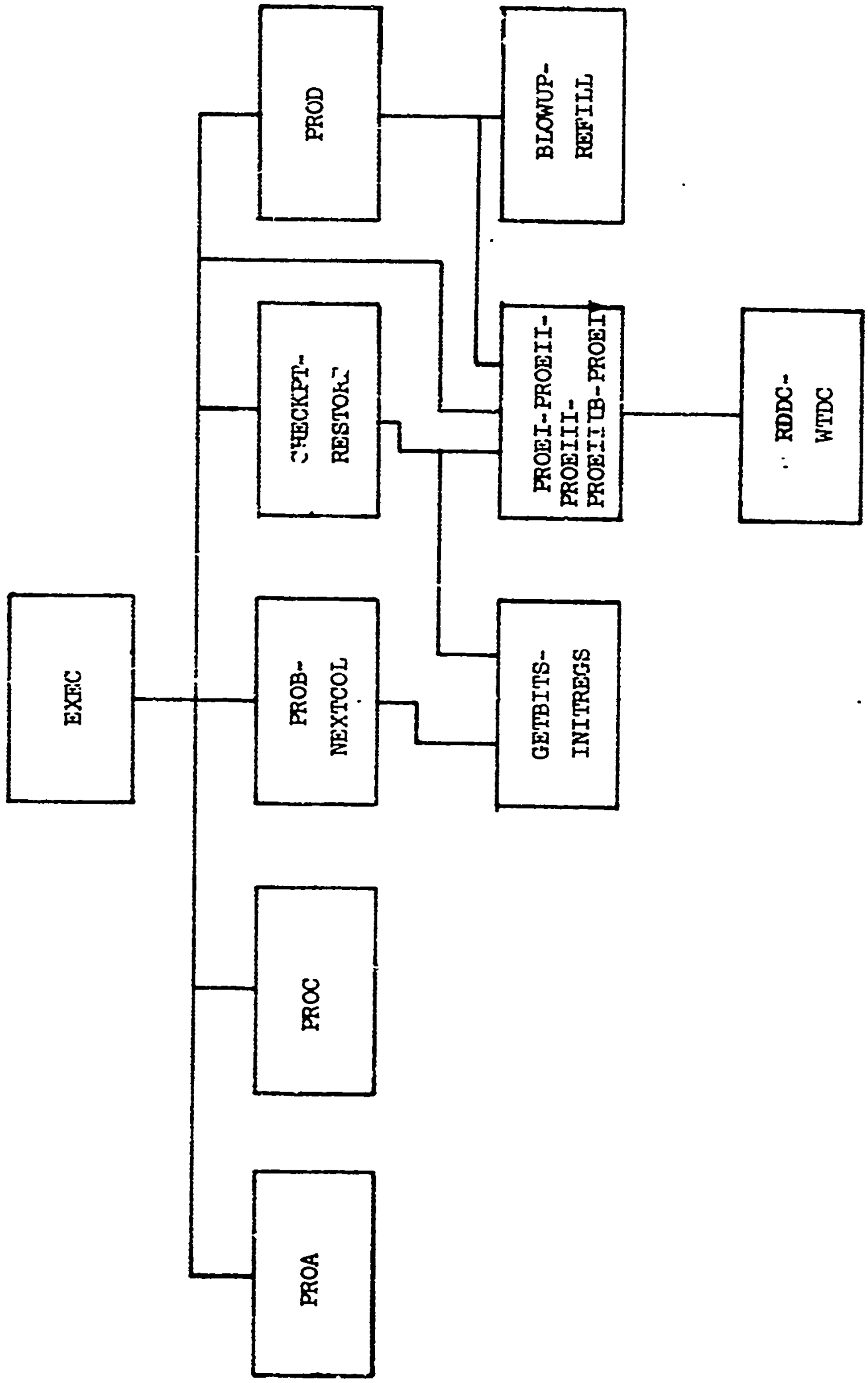


Figure 10 -- TASK 1 CALL TREE

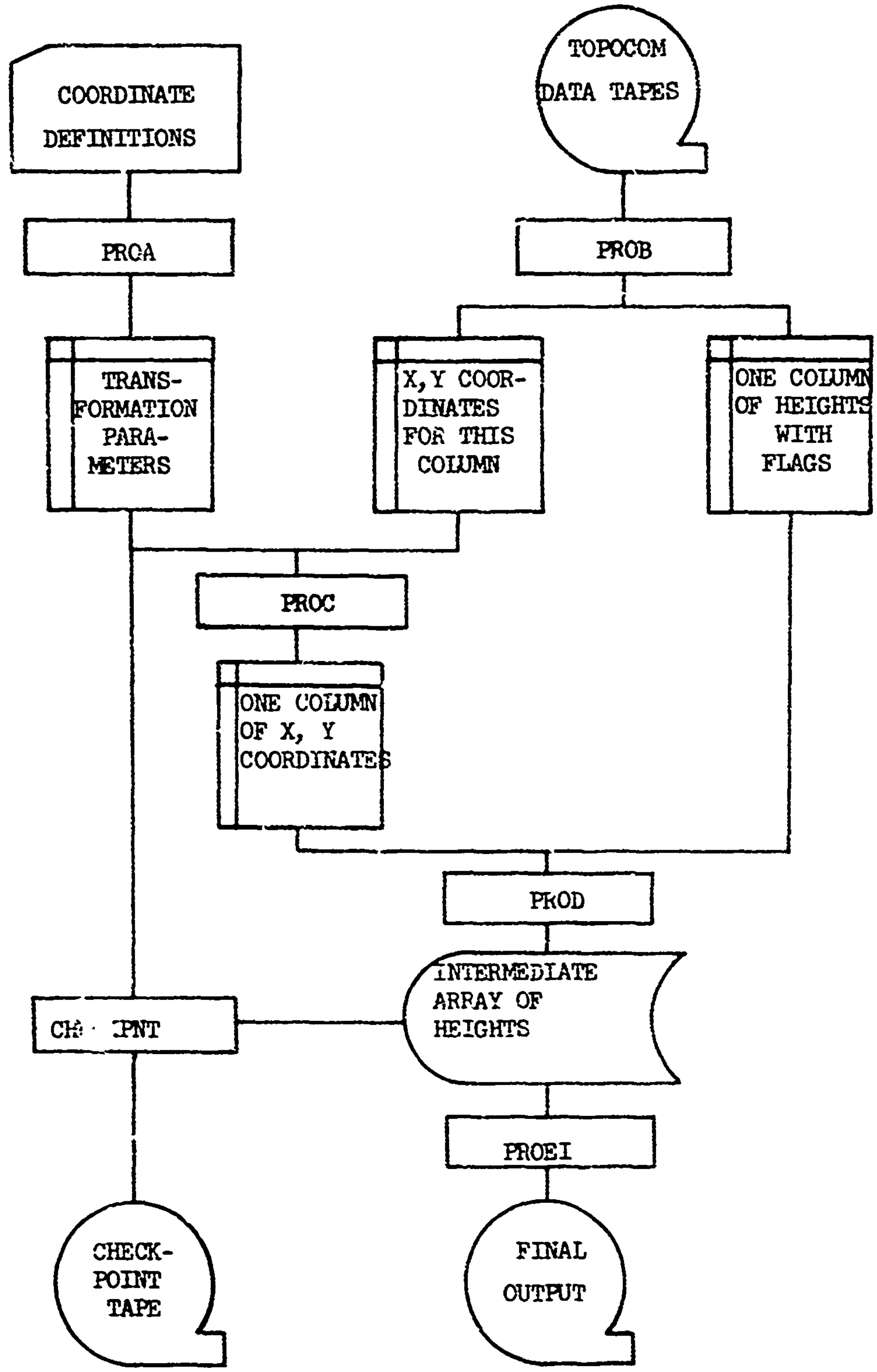


Figure 11 -- TASK 1 INTERFACE DIAGRAM

1.2 PROGRAM FLOW. The Task 1 program computes new coordinates for each point in the original data, and the data is arranged in columns (constant X') in the new coordinate system. The original data consists of six independent coordinate systems. If X,Y is a point in one of these coordinate systems, then the new coordinates X',Y' can be found from the transformation:

$$X' = X \cos \theta - Y \sin \theta + X_t$$

$$Y' = X \sin \theta + Y \cos \theta + Y_t$$

This is a rotation by an angle θ and a translation of an amount X_t in the X direction and Y_t in the Y direction. The values $\sin \theta$, $\cos \theta$, X_t, and Y_t are called transformation parameters and are computed for each of the six half-maps from west to east. As described in the detail documentation of subroutine PROA in Ref. 8, the angle between the left edge of the "current" map and the right edge of the "previous" map is found by the formula $\cos \theta = \cos (a-b) = \cos a \cos b + \sin a \sin b$, where cos a and sin a are the direction cosines of one edge and where cos b and sin b are the direction cosines of the other edge. (The direction cosines are found by dividing the X-component and the Y-component of any edge by the length of that edge as usual.) The rotation is applied to the current map's edge ($X_t = Y_t = 0$). Then the distance between the lower left corner of the current map (after rotation) and the lower right corner of the previous map (rotated earlier) is found by the formulas $X_t = X_a - X_b$ and $Y_t = Y_a - Y_b$ where (X_a, Y_a) and (X_b, Y_b) are the coordinates of the corners. The foregoing transformation, using these parameters, is applied to each map in turn.

A column of data (constant X) is read. The above equation is applied to each coordinate in this column to create a set of X',Y' coordinates. This set of coordinates will represent several (from 10 to 118) columns of constant Y'.

A sliding array of 120 columns is maintained on disc. Initially the leftmost column of this array corresponds to the zeroth column of the X',Y' coordinate system. The column of heights is placed into this array in the locations indicated by the X',Y' coordinetes.

The next column of data is read and the X',Y' coordinates are computed for it. Then the heights are stored into the disc array. After processing several columns of input data, the set of X',Y' coordinates will contain a contribution of the 120th column. The leftmost column is then reformatted and written onto the output tape. This column is now made to correspond to

the 120^{th} column. If i represents the lowest column currently represented in the disc array, then when data arrives for the $i+120^{th}$ column, the i^{th} column is written onto the output tape and the array "slides" one column.

After all of the input has been read, the disc array is emptied onto the output tape. Figure 12 is a macroflowchart of the Task 1 program.

The processing takes over twenty-four hours. Also the SIGMA-7 operating system cannot "know about"more than five different tapes at any given time. This program requires a total of ten tapes to be processed. Therefore, provision is made within the program to take a checkpoint after each input column is processed. To take a checkpoint all necessary variables (e.g., the transformation parameters) and the disc array are written onto a tape. Checkpoints are taken in response to the operator setting a sense switch or after a predetermined number of input records have been processed. When the program begins execution, a sense switch is checked to see if the checkpoint tape should be read instead of computing the transformation parameters.

1.3 PROGRAM OPERATION. This section serves as the User's Manual for the Task 1 program. It assumes that any arbitrary six sequential half maps are to be processed. There are two steps: preparation of the input deck, and operation of the computer.

1.3.1 Preparation of Input Deck. Figure 13 shows the program deck. It shows three subdecks and several control cards. Two of these three subdecks must be prepared particularly for those input tapes to be processed, and the third -- FORTRAN source program -- may have to be modified to adjust the disc array size.

To properly prepare the program deck the following data must be known for the input tapes to be processed: map name, corner coordinates, map sheet number, map sequence number, tape serial number. Table 1 shows this information for the six data tapes processed under this contract. The map sequence number is assigned from one to six and indicates the order from west to east that the tapes are to be processed. The data tape

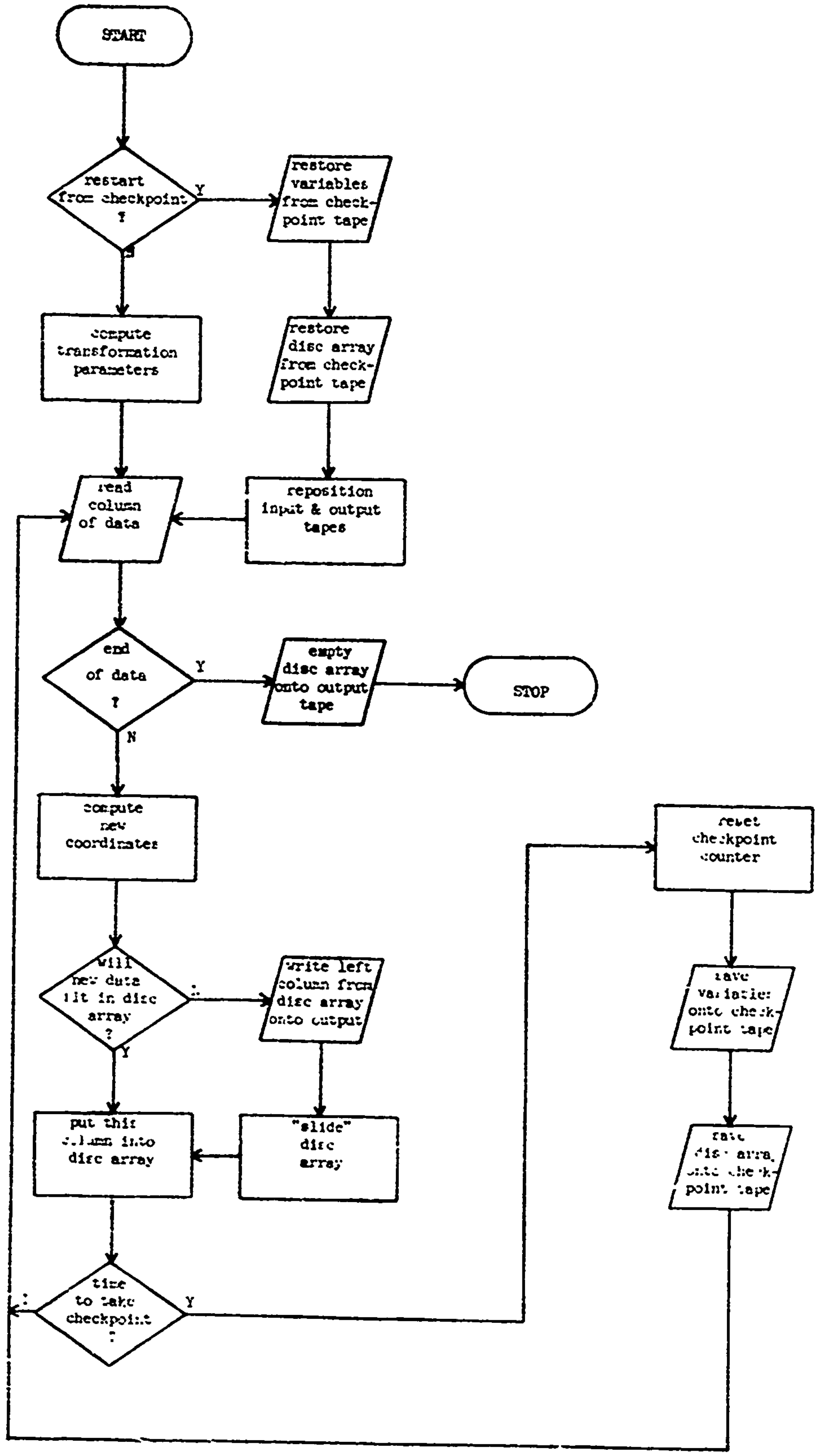


Figure 12 -- TASK 1 MACROFLOWCHART

serial numbers are assigned by the using computer center. The corner coordinates are supplied by TOPOCOM along with the data tapes. These coordinates are entered in Table 1 in clockwise order starting at the southwest corner. The coordinates are to be punched one card per map in accord with the FORTRAN format (4(2I.4,1)). Other data such as the map sequence number and map sheet name and possibly map sheet number may be punched on the right hand half of the data cards for convenience of the operator. Figure 14 shows the data deck used for the six data tapes processed.

The !ASSIGN cards are made up of the input !ASSIGN cards, output !ASSIGN cards, the disc file !ASSIGN card and the checkpoint !ASSIGN cards. The input !ASSIGN cards contain the data tape serial numbers from Table 1 or its equivalent. The sequence numbers assigned to the other !ASSIGN cards is up to the user. Figure 15 shows the input !ASSIGN cards; Figure 16 shows the output !ASSIGN cards, Figure 17 shows the disc !ASSIGN card; and Figure 18 shows the checkpoint !ASSIGN cards. At no time may there be more than five !ASSIGN cards which refer to tapes. For this reason the deck must be modified periodically throughout the run. The set of !ASSIGN cards will usually consist of the following:

- the !ASSIGN card for the current input tape
- the !ASSIGN card for the next input tape
- the !ASSIGN card for the current output tape
- the !ASSIGN card for the disc (This doesn't count in the limit of five.)
- the two !ASSIGN cards for the checkpoint tapes.

Once an input tape has been processed and the program switches to the next input tape, a checkpoint should be taken and the program stopped. The deck may now be rearranged by replacing the !ASSIGN card for the just completed input tape by the !ASSIGN card for the tape to follow the one now being processed. If the output tape is near completion the !ASSIGN card for the second output tape should be placed in the deck instead of the !ASSIGN card for a new input tape.

TABLE 1 -- PARAMETERS FOR THE WARREN, WILLIAMSPORT, AND SCRANTON MAPS

MAP NAME	CORNER COORDINATES				MAP SHEET NUMBER	MAP SEQUENCE NUMBER	TAPE SERIAL NUMBER
Warren	0174,0125	0193,1872	1499,1864	1499,0116	NK17-9W	1	0005
Warren	0147,0118	0147,1865	1452,1872	1472,0124	NK17-9E	2	0010
Williamsport	0175,0125	0194,1873	1499,1866	1499,0116	NK18-7W	3	0040
Williamsport	0162,0117	0162,1866	1467,1874	1487,0126	NK18-7E	4	0039
Scranton	0117,0126	0137,1872	1442,1864	1441,0118	NK18-8W	5	0042
Scranton	0200,0081	0200,1827	1506,1834	1527,0087	NK18-8E	6	0041

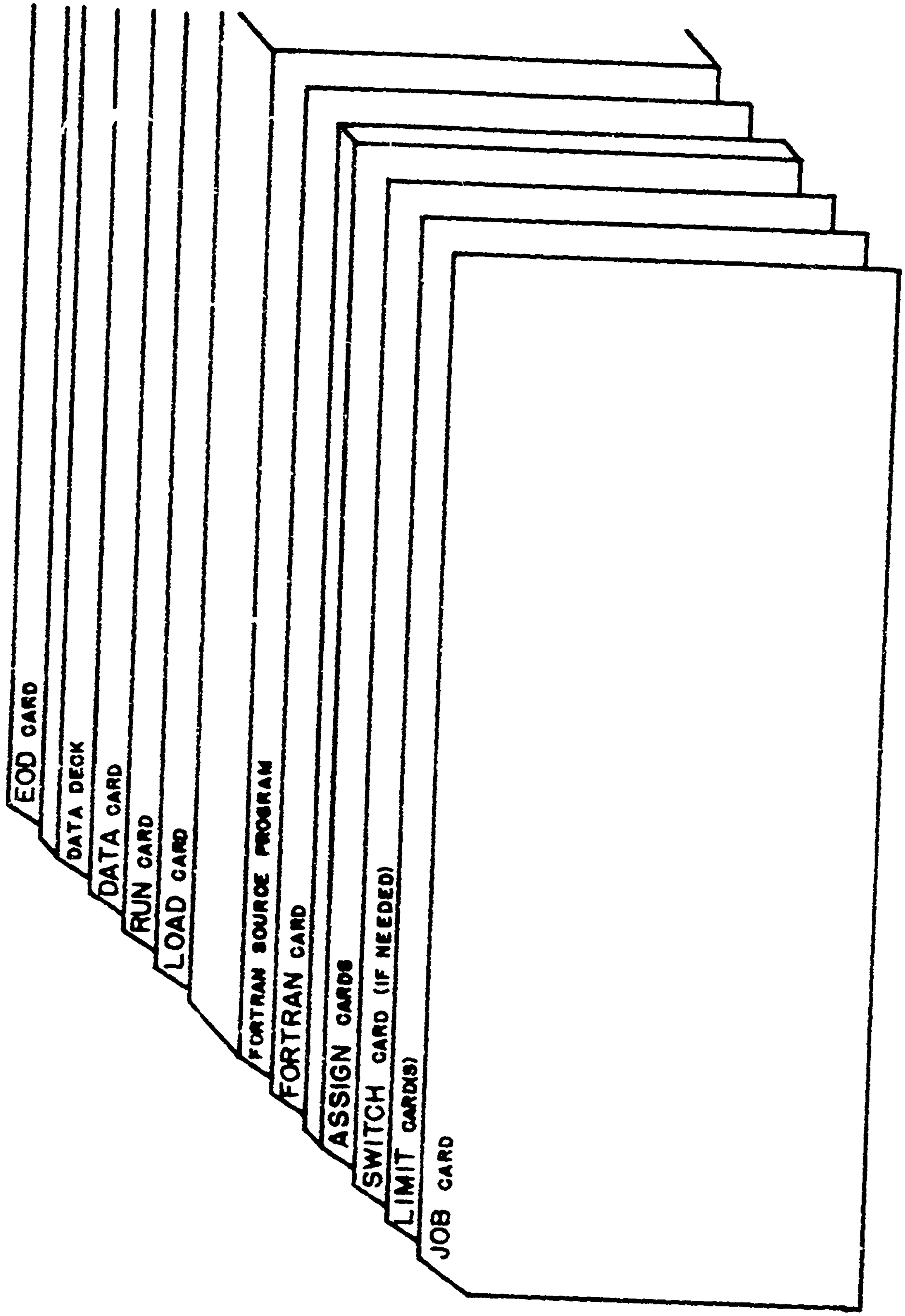


Figure 13 -- TASK 1 PROGRAM DECK

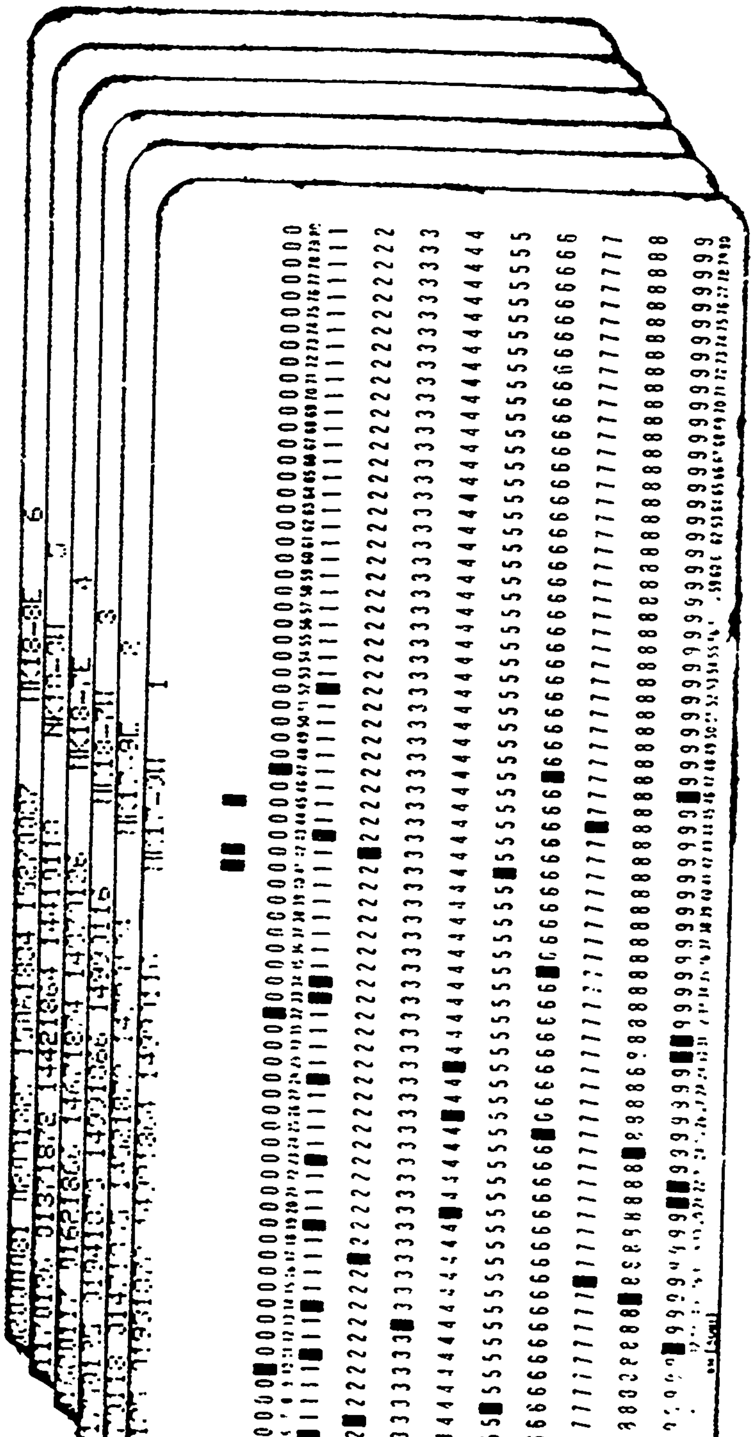


Figure 14 -- DATA DECK FOR WARREN, WILLIAMSPORT AND SCRANTON

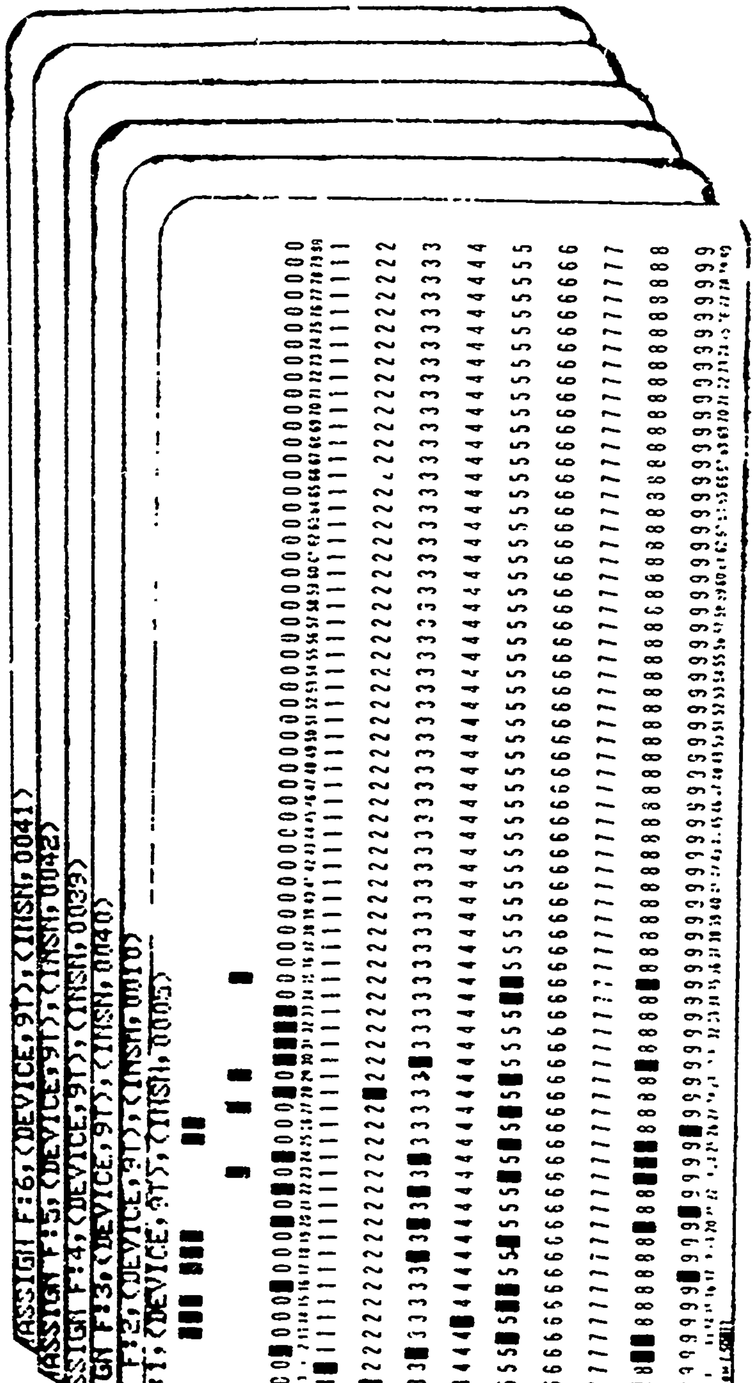


Figure 15 -- TASK 1 INPUT !ASSIGN CARDS

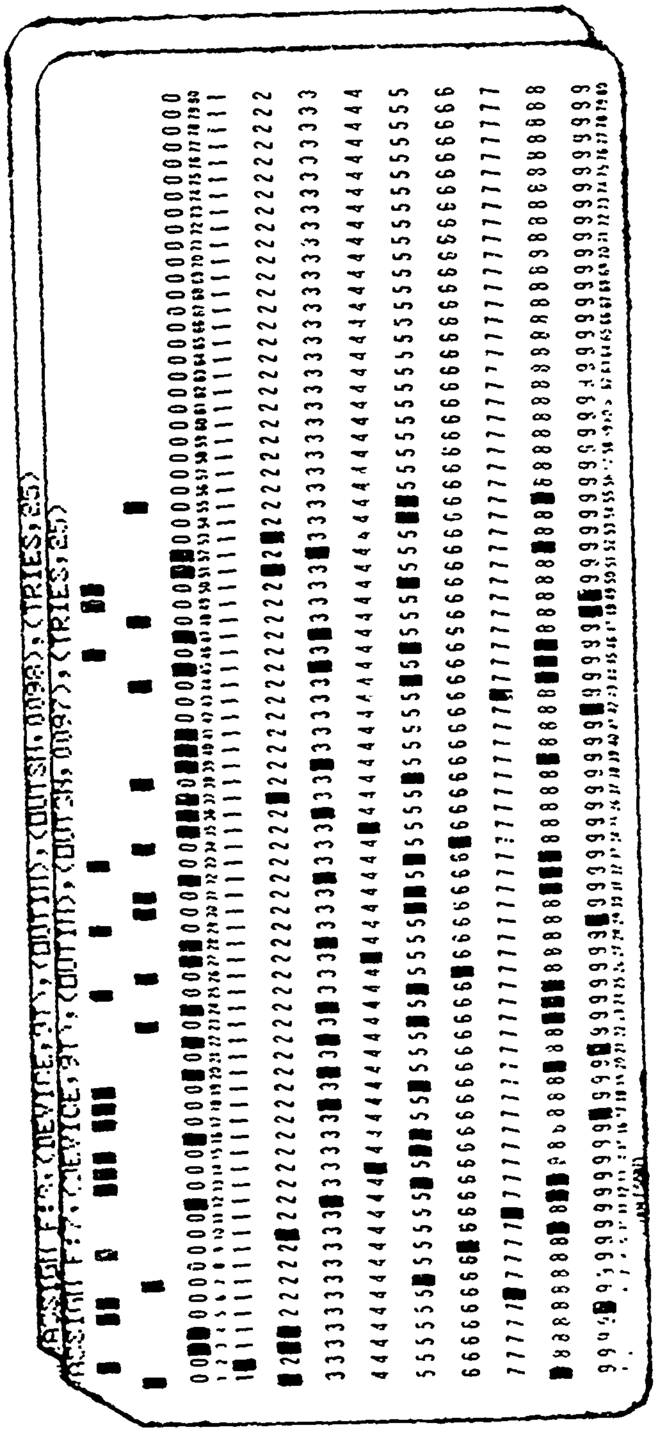


Figure 16 -- TASK 1 OUTPUT !ASSIGN CARDS

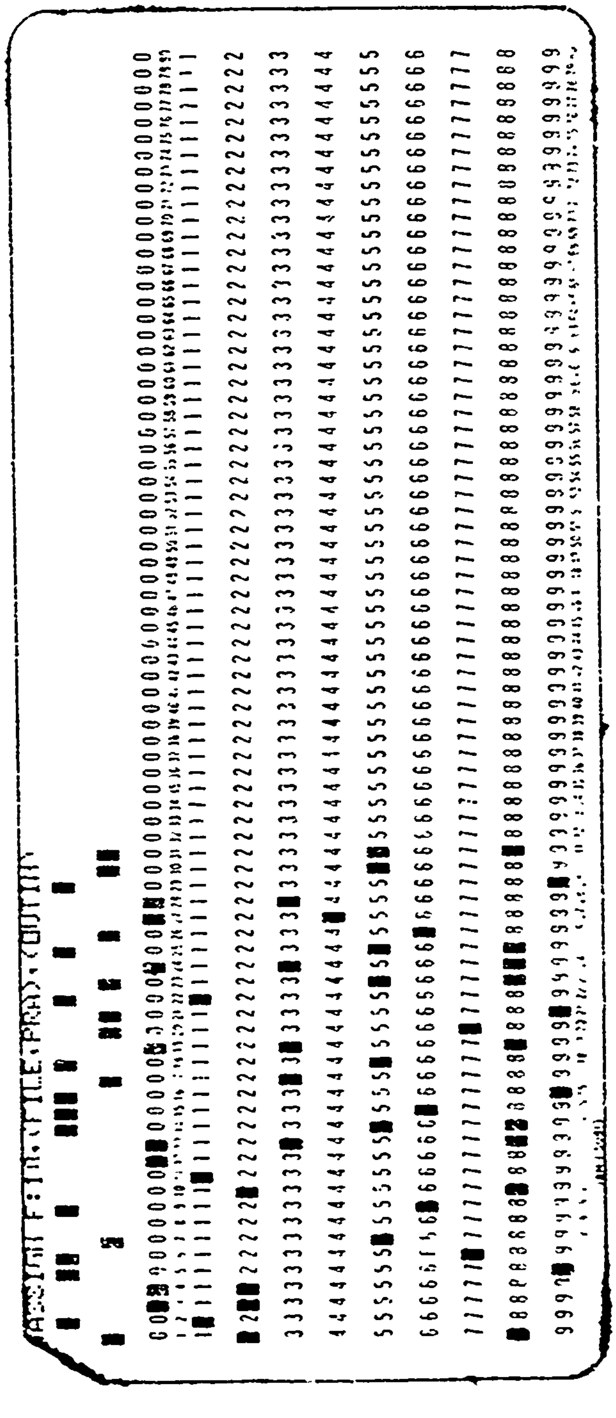


Figure 17 -- TASK 1 DISC :ASSIGN CARDS

1.3.2 Computer Operation. This program operates as a batch program under BPM. There are three circumstances which require special operator action: inital starting of a program, taking a checkpoint, and restoring from a checkpoint.

1.3.2.1 Initial Start. The deck should be arranged as shown in Figure 13 with the following changes:

- there should be no switch card
- the !ASSIGN cards should consist of:
  - the !ASSIGN card for the first input tape
  - the !ASSIGN card for the second input tape
  - the !ASSIGN card for the first output tape
  - the !ASSIGN card for the disc
  - the !ASSIGN cards for both checkpoints tapes

The data cards should be properly prepared and in proper order.

Start the program in the normal BPM manner. Mount the tapes on the drives in response to the messages typed by the operating system. When an input tape has been completely processed it will be rewound and a request made to mount the next input tape. Soon after the second input tape has begun processing it is advisable to take a checkpoint and stop the program and change the !ASSIGN card as indicated in Section III.1.3.1 above.

1.3.2.2 Taking a Checkpoint. The taking of a checkpoint may be initiated by the operator or may be done periodically by the program. Ideally this period should be between one half hour and one hour. Table 2 shows the number of records to be processed for approximately one half hour checkpoints versus which input is to be processed. Initially this period is 600 records, however, when the last tape is being processed this period is merely 225 records. To change this period the 40th card of the main programs must be changed. This card currently reads

```
IF (KOUNTER.GE.225) GO TO 40
```

The number 225 should be changed to the desired checkpoint interval.

TABLE 2 -- CHECKPOINT INTERNAL	
MAP NUMBER	APPROXIMATE NUMBER OF RECORDS PER HALF HOUR
1	600
2	525
3	450
4	375
5	300
6	225

Alternatively the operator may initiate checkpoints by setting the BPM virtual sense switch 2 to the ON state, by the procedure of Section III.2.3.4.3. In either case when a checkpoint is being taken the following procedure should be followed:

- The typewriter will type
  ENTER CHECKPT,RECORD SAVE TAPE NUMBER

  At this point the typewriter lamp will light. The operator may now make any notations he desires such as the serial number of the checkpoint next to be used. The checkpoint tapes are automatically alternated beginning with the first checkpoint tape (i.e., that tape indicated by the !ASSIGN for F:11). It is also advisable to note the time of day.

- The new line key should be pressed. The system will now begin writing the information on the current checkpoint tape. It is possible that the system will type a !!MOUNT message for the checkpoint tape -- this should be responded to accordingly.

- If it is desired to continue processing after this checkpoint has been taken, the BPM virtual sense switch 2 should now be set OFF, as described in Section III.2.3.4.3.

If sense switch 2 was set OFF as above the program will now continue processing; otherwise, the program will come to an orderly halt.

1.3.2.3 Restoring From a Checkpoint. To restart from a checkpoint the job is started as described above in Section III.1.3.1 with one major exception. The !SWITCH card as indicated in Figure 13 should be inserted as follows:

```
!SWITCH (SET,1)
```

The program will request that the checkpoint tape be mounted first. The checkpoint tape mounted in response to this request should be that tape on which the desired checkpoint was taken. The serial number indicated in the !!MOUNT message may be wrong. It is therefore very important that when checkpoints are taken (see Section III.1.3.2.2 above) that the serial number of the tape on which the checkpoint is being taken is recorded along with the time of day and any other pertinent information.

2. COMPRESS TERRAIN

This section describes the program that compresses the terrain data by computing coefficients of a polynomial. This program accepts gridded height values from tape stored as columns of constant X. It also reads cards which specify the value of those parameters listed in Section II.2.6. One of these parameters -- the basis functions -- cannot easily be specified by cards which are read at execution time. Instead a subroutine written in FORTRAN (or some other language acceptable to the object computer) must be included in the program deck to specify the basis functions. The output of this program is a tape containing one record for each region. (The definition of a region is given in Section II.2.1.) The first two words of each record will be the X and Y numbers for the region (the lower-left region is assigned the number 0,0) followed by the coefficients packed two per word and scaled as indicated by the parameter cards.

The program structure is now described, followed by a description of the program flow. Finally a discussion of program operation, including preparation of and limitations to the parameters, is presented.

2.1 PROGRAM STRUCTURE. The program consists of two components: the computational section and the control section. The computational section processes each region individually. Since it is desirable that there be an overlap of the data processed for each region, each height value may be

processed by the computational section more than one time. The control section reads the height values and gives them to the computational section along with the proper control parameters. It also collects the output of the computational section, reformats it, and writes it onto the output tape. Figure 19 shows this structure. The names inside the boxes are subroutine names. Details of the individual subroutines are described in Ref. 9.

2.1.1 Computational Section. For each region a coordinate system is defined thus:

Region boundaries are (0,0), (0,1), (1,1), and (1,0)

Group boundaries are (-g,-g), (-g,1+g), (1+g,1+g), and (1+g,-g)

where $0 \leq g \leq \frac{1}{2}$

If the region size is N_r grid elements on a side and the group size is N_g grid elements, then

$$g = \frac{N_g - N_r}{2\,N_r} \qquad (1)$$

Figure 20 shows this coordinate system.

Let x_w represent the set of all x-coordinates of elements included in a group

and y_w represent the set of all y-coordinates of elements included in a group

then $-g \leq \begin{pmatrix} x_w \\ y_w \end{pmatrix} \leq 1+g$

and set X equals the set Y.

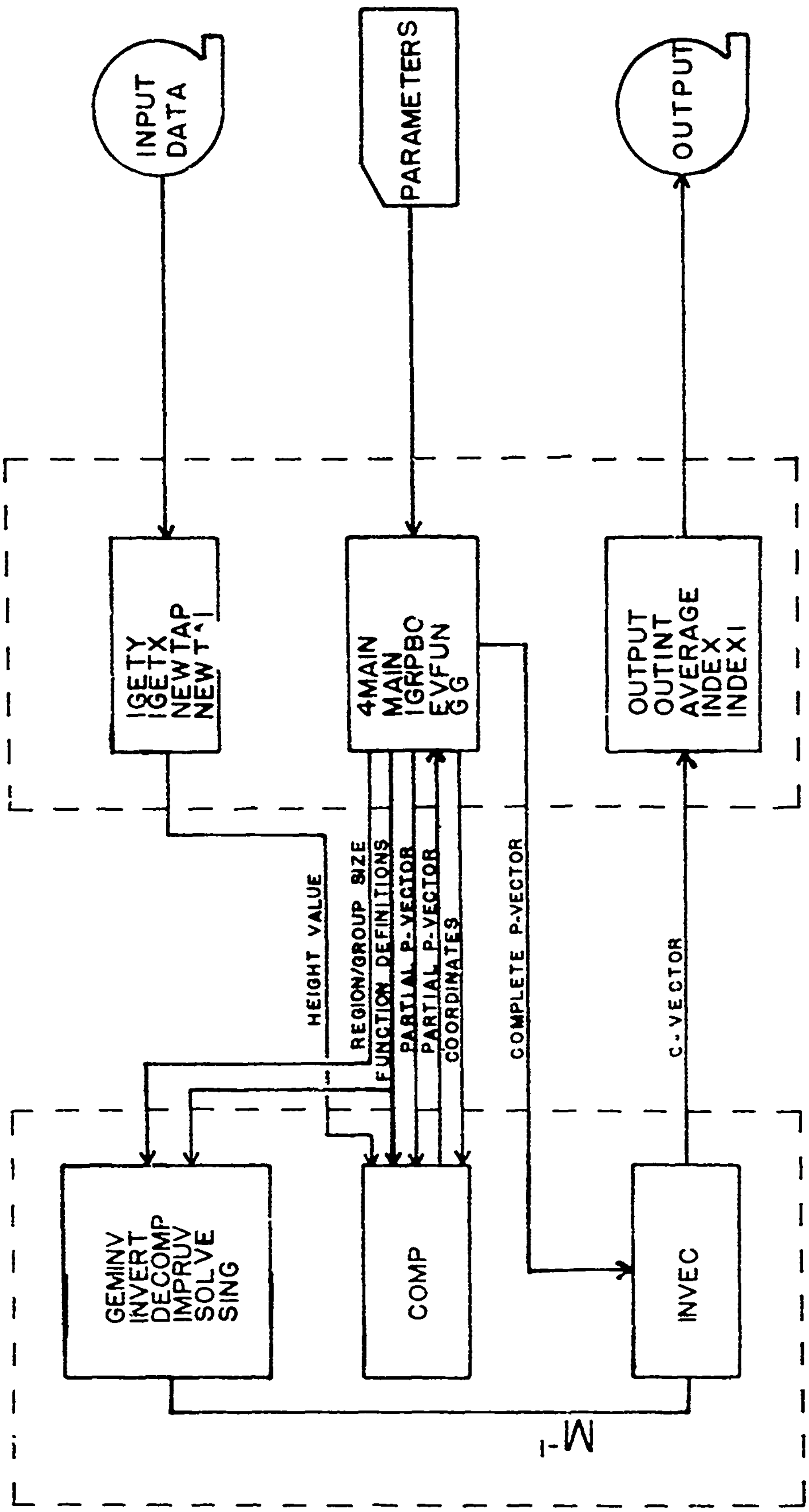


Figure 19 -- TASK 2 PROGRAM STRUCTURE

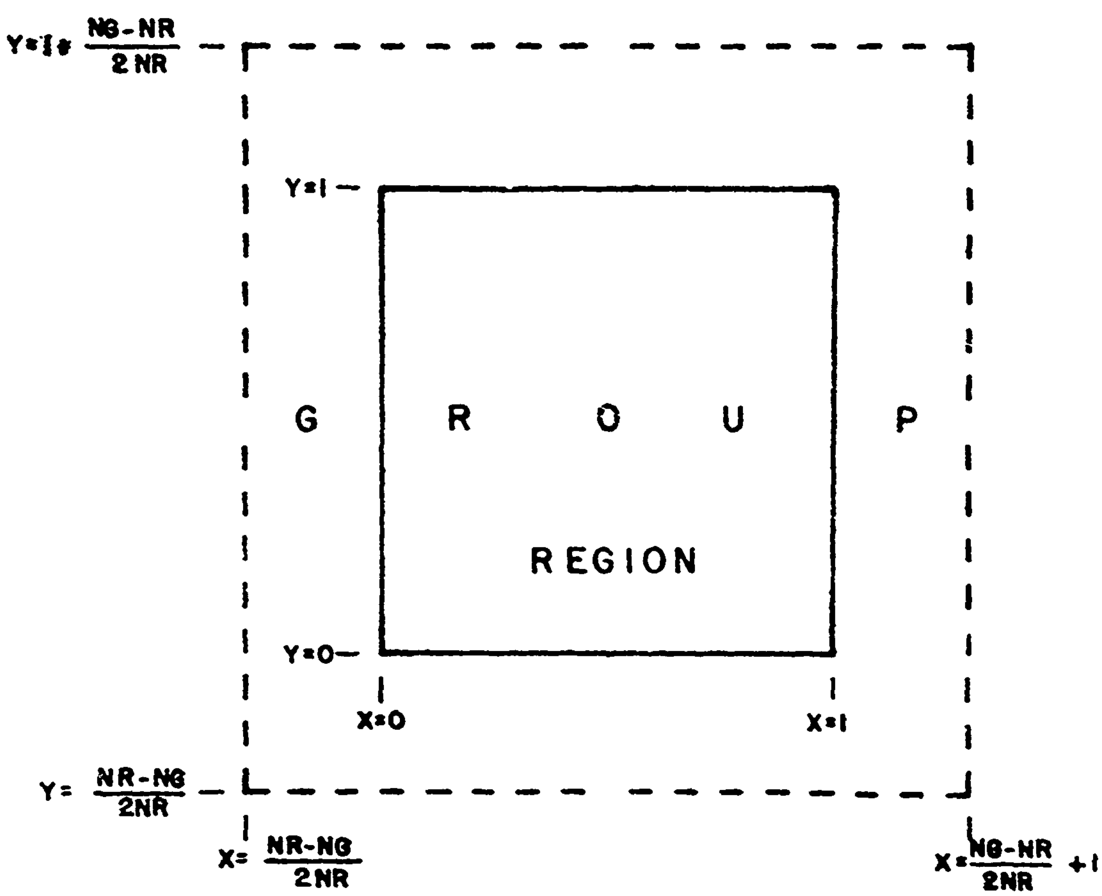


Figure 20 -- LOCAL COORDINATE SYSTEM

Let $z(x_w, y_w)$ represent the actual height values at the point x_w, y_w and $h(x_w, y_w)$ represent the approximated height values at the same point.

It is desired to compute a vector of coefficients C such that

$$\sum_w \left(z(x_w, y_w) - h(x_w, y_w)\right)^2 \tag{2}$$

is minimized.

In Section II.2 $h(x,y)$ was defined by

$$h(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{i,j} \; G_i(x) \; G_j(y) \tag{3}$$

Substituting equation (3) into formula (2), differentiating the result with respect to c_{IJ}, and setting this result equal to zero yields the following equation:

$$\sum_w \left[\sum_{i=1}^{n} \sum_{j=1}^{n} G_i(x_w) G_j(y_w) G_I(x_w) G_J(y_w) c_{ij}\right] = \sum_w G_I(x_w) G_J(y_w) \; z(x_w, y_w) \tag{4}$$

Equation (4) represents n^2 simultaneous equations -- one for each possible value of I and J. The vector of n^2 c's is the set of coefficients which will minimize formula (2).

This set of equations can be written in matrix form MC = P,

where M is an $n^2 \times n^2$ matrix with the elements m_{Tt}; $T = n(I-1) + J$, $t = n(i-1) + j$, and (5)

$$m_{Tt} = \sum_w G_i(x_w) \; G_I(x_w) \; G_j(y_w) \; G_J(y_w) \tag{6}$$

C is the n^2 element vector defined by,

$$C = (c_{11}, c_{12}, \ldots c_{nn}) \tag{7}$$

P is the n^2 element vector with the elements p_T defined by,

$$p_T = \sum_w G_I(x_w) \; G_J(y_w) \; z(x_w, y_w) \tag{8}$$

Finding C requires inverting M and multiplying the P, thus

$$C = M^{-1} P \tag{9}$$

For the Lagrange 2-2 polynomial the number of functions $n = 6$. Therefore, the matrix M is 36 x 36. Standard matrix inversion packages tend to produce poorer results the larger the matrix. Unless one uses double precision arithmetic or an iterative improvement (see Ref. 7) even a 6 x 6 matrix can be inverted to only one significant digit accuracy. The use of both double precision and iterative improvement (this presents special difficulties) probably would not be of any help. Programs to invert large matrices (e.g., linear programming packages) are tailored to matrices with particular properties (e.g., sparceness). Fortunately the matrix M has a special property which is now presented*.

The matrix M is defined above by equation (6). Each element of M is a sum over all grid points w within the group being processed. Since the grid points are uniformly spaced the set X (and the set Y which is equal to X) is the same for all groups and is defined thus:

*Acknowledgement is hereby given to Dr. F. J. Murray of Duke University who pointed out the property of the matrix M which makes the computation of M inverse so computationelly simple.

$$x_w, y_w = \frac{k}{N_r} - g, \frac{\ell}{N_r} - g \quad \text{for all } k = 0, N_g \text{ and all } \ell=0, N_g \qquad (10)$$

where N_r is the number of grid points along a side of a region,
where N_g is the number of grid points along a side of a group,
and g is defined by equation (1) above. The summation over w now becomes summation over k and l; equation (6) may, therefore, be rewritten

$$m_{Tt} = \sum_{k=0}^{N_g} \sum_{l=0}^{N_g} G_i(x_k)\, G_I(x_k)\, G_j(y_l)\, G_J(y_l) \qquad (11)$$

Two matrices, a and b, each nxn are defined to have the following elements:

$$a_{Ii} = \sum_{k=0}^{N_g} G_i(x_k)\, G_I(x_k) \qquad (12)$$

$$b_{Jj} = \sum_{l=0}^{N_g} G_j(y_l)\, G_J(y_l) \qquad (13)$$

(It should be pointed out here that a and b are equal, but this is not necessary for the argument that follows and treating them separately makes this argument easier to understand. For actual computation, however, only one matrix will be generated.)

Let $\alpha = a^{-1}$ with the elements α_{iI}

and $\beta = b^{-1}$ with the elements β_{jJ}

By the definition of matrix multiplication and matrix inverse it follows that

$$\sum_{i} a_{Ii}\,\alpha_{iK} = \delta_{IK} \tag{14}$$

and

$$\sum_{j} b_{Jj}\,\beta_{jL} = \delta_{JL} \tag{15}$$

The δ_{op} is known as the Kroneker delta and is equal to one if and only if $o = p$ and is otherwise equal to zero. (The matrix M is the Kroneker product of the matrices a and b.)

Consider the matrix A with the elements

$$A_{sS} = \alpha_{kK}\,\beta_{lL}$$

where

$$\begin{aligned} s &= n(k-1) + l \\ S &= n(K-1) + k \end{aligned} \tag{16}$$

(i.e., the Kroneker product of α and β.)

Then

$$\sum_{t} m_{Tt}\,A_{tS} = \sum_{t=1}^{n^2} a_{Ii}\,b_{Jj}\,\alpha_{iK}\,\beta_{jL} \tag{17}$$

where $t = n(i-1) + j$

$$= \sum_{i=1}^{n}\sum_{j=1}^{n} a_{Ii}\,b_{Jj}\,\alpha_{iK}\,\beta_{jL} \tag{18}$$

$$= \sum_{i=1}^{n} a_{Ii}\,\alpha_{iK} \sum_{j=1}^{n} b_{Jj}\,\beta_{jL} \tag{19}$$

$$= \delta_{IK}\quad \delta_{JL} \tag{20}$$

This is equal to one if and only if I=K and J=L, and is equal to zero otherwise. Defining $T = n(I-1) + J$ as before and S as above

$$\sum_t M_{Tt} A_{tS} = \delta_{TS} \tag{21}$$

Therefore

$$A = M^{-1}$$

To compute M^{-1} it is only necessary to invert the $n \times n$ matrix a to produce α and generate the Kroneker product of α with itself.

The computational section consists of three modules: one to compute M^{-1}, one to compute the vector P, and one to compute the vector C. The interface with these modules is thus: (See Figure 19)

Compute M^{-1}

- input:

  Function definitions
  Region size
  Group size

- output:

  The matrix M^{-1}

Compute P

- input:

  A height value
  Coordinates of value with respect to region being processed.
  Partially computed P vector
  Function definitions

Compute P cont'd.

- output:

  Partially computed P vector (with the input height's contribution added)

Compute C

- input:

  The P vector for this region
  The matrix M^{-1}

- output:

  The vector C

2.1.2 Control Section. This section is the interface between the computational section and the outside world. It consists of three modules: input, executive, and output. The input module is the interface with the input tape(s); it furnishes a height value upon request. The executive module, controls the flow of data and control through the other modules; it calls for data from the input, computes relative coordinates, calls upon the appropriate modules of the computational section, and calls upon the output module when required. The output module is the interface with the output tape; it takes the C-vector from the computational section, averages it with the C-vectors of neighboring regions as required, reformats the resulting average, and outputs it.

Details of the modules in the control section are in Ref. 9 under the discussion of the following routines:

input module:	IGETX and IGETY
executive module:	4MAIN and MAIN
output module:	AVERAGE, INDEX, and OUTINT-OUTPUT

A general description of the control section is in the discussion of program flow below.

2.2 PROGRAM FLOW. The program begins by reading and interpreting the parameter cards. Storage is then allocated and the module within the computational section is called upon to compute M^{-1}. The program is then ready to process the input data -- one height value at a time. Figure 21 is a macro flowchart of the program.

The data is organized on the input by columns of constant x. The processing is, therefore, done by columns. The problem area is also organized into columns of regions with their associated groups. The main loop of the program reads and processes a column of data. Every few columns (this interval is the region size) a pass is made through the columns of P-vectors which have been completed to turn them into C-vectors. The output module then averages these C-vectors with the previous column and outputs the previous column.

On each pass through the main loop sense switch 1 is examined. If it is ON the checkpoint subroutine is called upon to write all internal variables -- including the partially completed P-vectors and partially averaged C-vectors -- onto a tape. If sense switch 2 is also ON the program continues, otherwise the program stops. The program may be restarted by setting sense switch 2 ON by means of a !SWITCH card when starting the program.

The program continues (the checkpoint taking has no effect) by reading the next column. The program processes every column of regions in the defined problem area whether or not there is any data for the total area or not. The left most or the right most column may not have data associated with it. If a request is made of the input module (subroutine IGETX) to fetch a column of data which exists within the problem area, but for which there is not any data, a null column indication is returned instead. The program then sets the P-vectors for all groups in this region column to null and proceeds as if all data for the current region column has been processed.

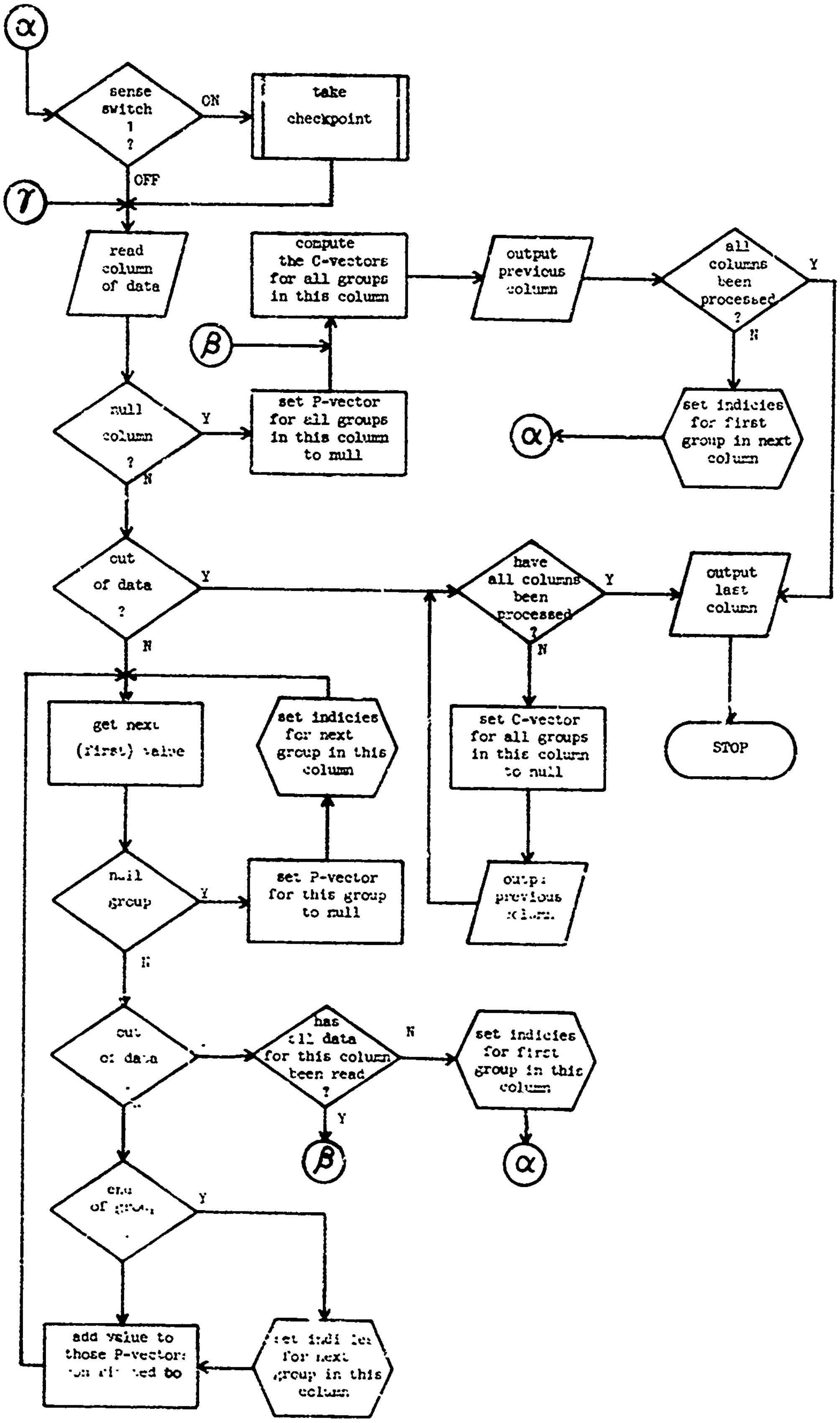


Figure 21 -- TASK 2 MACROFLOWCHART
(Sheet 2 of 2)

If data is returned from IGETZ, a pass is then made through the data. Again, processing is for every region (with its associated group) in the problem area. At the bottom or top there may be groups that lie outside the area covered by the input data. If the currently processed group has no data associated with it, the input module (subroutine IGETY) will return a null indicator instead of data. In this case the P-vector for this group is set to null and the next group is made the current group.

Each height value may contribute to more than one group. In addition to the current group, a height value may contribute to the next higher and (in cases of maximum overlap) the second higher group. It may also contribute to the next right groups to the current and next higher group(s). (In cases of maximum overlap the second right groups are handled by special logic in the input module.) The program maintains partial P-vectors for two columns of groups. The partial P-vectors are fed along with the heights and the relative coordinates of the height to the appropriate module (subroutine COMP) in the computational section. This is done for each group to which the current height contributes.

If all heights in the current data column which contribute to the current group have been processed, the next higher group becomes the current group. After all height data in the current column has been processed, a check is made to see if all columns of data which contribute to the current group column have been processed. If not, the first group in the column is again made the current group and the loop continues by checking sense switch 1.

If all data contributing to the current group column has been processed, the P-vectors are converted one at a time to C-vectors. These C-vectors are then fed to the output module which averages appropriate components with the previous column. This previous column is then reformatted and output. If all region (group) columns in the problem area have been processed, the last column is reformatted and output, and the programs stops. Otherwise processing continues by checking sense switch 1 to see if a checkpoint should be taken.

If, when the input module is called upon to fetch the next data column, it is found that there is no more data, the remaining columns are set to null. After all remaining columns in the problem area have been accounted for, the program stops.

2.3 PROGRAM OPERATION. This section serves as an user/operator manual for the Task 2 program. It is divided into four subsections: definition of the basis functions, preparation of the parameter cards, calculation of required storage, and operation of the computer.

2.3.1 Definition of the Basis Functions. The approximation of height $h(x,y)$ has been defined above as

$$h(x,y) = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}\, G_i(x)\, G_j(y).$$

The G's are the basis functions. Provision is made for the definition of any desired set of basis functions -- up to ten functions.

The Task 2 program includes a definition of the normalized Lagrange 2-2 polynomials. To define another set the subroutine GG must be replaced by a subroutine with the following characteristics:

- It must be equivalent to the following FORTRAN subroutine

```
SUBROUTINE GG(X,V)
GLOBAL N
DATA N/ the-number-of-functions/
DIMENSION V (the-number-of-functions)
coding as necessary to store intoV(9i) the value G_i(X)
RETURN
END
```

- The argument X is a real value between 10.5 and 1.5. The region boundaries are at X=0.0 and X=1.0.

- The functions should be normalized so that for all G_i and all X such that $-0.5 \leq X \leq 1.5$, $-1.0 \leq G_i(X) \leq 1.0$.

2.3.2. <u>Preparation of the Parameter Cards</u>. There are six types of parameter cards, each specifying one of the following:

- Region/Group Size
- Problem Area Size and Boundaries
- Input Unit(s)
- Coefficient Scaling
- Output Unit
- Coefficient Sharing (two cards required)

Figure 22 shows the layout of each of these cards.

2.3.2.1 Region/Group Size Specification.

Card Type Code:	01	
Variables:	NR	The number of data points along side of a region.
	NG	The number of data points along side of of a group.
Restriction:	NR $\leq$ NG $\leq$ 2NR	
FORTRAN FORMAT:	(I2,4X,I4,4X,I4)	

2.3.2.2 Problem Area Size and Boundaries. The problem area is rectangular and is specified in terms of the data base's coordinate system. The units are data grid points.

Card Type Code: 02

Variables:
- XMIN The minimum x-coordinates of the problem area
- YMIN The minimum y-coordinate of the problem area
- XMAX The maximum x-coordinate of the problem area
- YMAX The maximum y-coordinate of the problem area

Restriction:
XMIN < XMAX
YMIN < YMAX

FORTRAN FORMAT: (I2,4(6X,I6)

2.3.2.3 Input Unit(s).

Card Type Code: 03

Variables:
- NOTAPE The number of input tapes to be read
- UNITS The actual logical unit number for each input tape

Restriction: NOTAPE ≤ 10

There must an !ASSIGN card for each unit number specified.

FORTRAN FORMAT: (I2,7X,10(I2,1X)

2.3.2.4 Coefficient Scaling.

Card Type Code: 04

Varibles:
- SIGMA The power of two represented by the most signficant bit in the outputed coefficients (exclusive of sign).
- TAU The power of two, represented by the least significant bit in the outputed coefficients.

Restriction: SIGMA-TAU+1 ≤ 15

FORTRAN FORMAT: (I2,7X,I3,5X,I3)

```
01 NR=xxxx NG=xxxx
02 XMIN=xxxxxx YMIN=xxxxxx XMAX=xxxxxx YMAX=xxxxxx
03 NOTAPE=xx UNITS=xx,xx,xx,xx,xx,xx,xx,xx,xx,xx
04 SIGMA=xxx TAU=xxx
05 OUTPUT=xx
06 PMAX=xx QMAX=xx ELEMENTS=x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x
06                         x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x x,x
```

Figure .2 -- TASK 2 PARAMETER CA'DS FORMAT

2.3.2.5 **Output Unit.**

Card Type Code:	05	
Variables:	OUTPUT	The actual logical unit number for the output tape.
Restriction:	There must an !ASSIGN card for this unit. This unit may not be one of the above used for input.	
FORTRAN FORMAT:	(I2,8X,I2)	

2.3.2.6 Coefficient Sharing. If the number of functions is denoted by N., then there are N^2 coefficients. They are allocated to storage as though they were in the following FORTRAN array:

```
CVECTOR (0:N/2-1,0:1,0:N/2-1,0:1)
```

and accessed as CVECTOR (Q,J,P,I) in the normal FORTRAN manner. The indices I,J indicate which corner by I=X and J=Y. The indices indicate which derivative for Lagrange n-n polynomials (n=N/2-1). For other basis functions it merely refers to one of the $N^2/4$ possible coefficients for each corner. Two cards are required to specify which of the $N^2/4$ coefficients are to be averaged and shared with the neighboring regions.

Card Type Code:	06	
Variables:	PMAX	Equal to N/2-1
	QMAX	Equal to N/2-1
	ELEMENTS	The actual P,Q pairs for the elements to be averaged.
Restriction:	Since it is possible that there be 25 pairs (for N=10) a second card with only the type code must be included if less than 13 elements are to be averaged.	
FORTRAN FORMAT:	(I2,6X,I2,13(I1,1X,I1)) This FORMAT is used to read each card individually.	

2.3.3 Calculation of Storage Required. The number of words of core storage required for internal arrays is calculated by the following formula:

$$1 + N(NG+1) + \frac{N^2(N^2+1)}{2} + \frac{N^2\ (YMAX-YMIN)}{NR}$$

where N is the number of functions
NG is the size of a group
NR. is the size of a region
YMAX is the maximum y-coordinate of the problem area
YMIN is the minimum y-coordinate of the problem area

There are 13,345 words available. Therefore, the above formula must be a number less than or equal to 13,345.

2.3.4 Computer Operation. The deck set up is shown in Figure 23. The !ASSIGN cards are used as follows:

- F:1 is the first input tape
- F:2 is the second input tape
- F:33 is the output tape
- M:CK is the checkpoint tape.

The option PERM is included on the !LOAD card so that reloading of the binary deck is not necessary for subsequent restarts from checkpoint. If the binary deck still exists as the user file Task 2 the binary deck beginning at the !LOAD and ending with the !EOB card need not be included in the deck.

The Task 2 program requires very little operator intervention. The following circumstances, however, require operator intervention: initial start up, taking a checkpoint and continuing, taking a checkpoint and terminating, switching input tapes, and starting from a checkpoint.

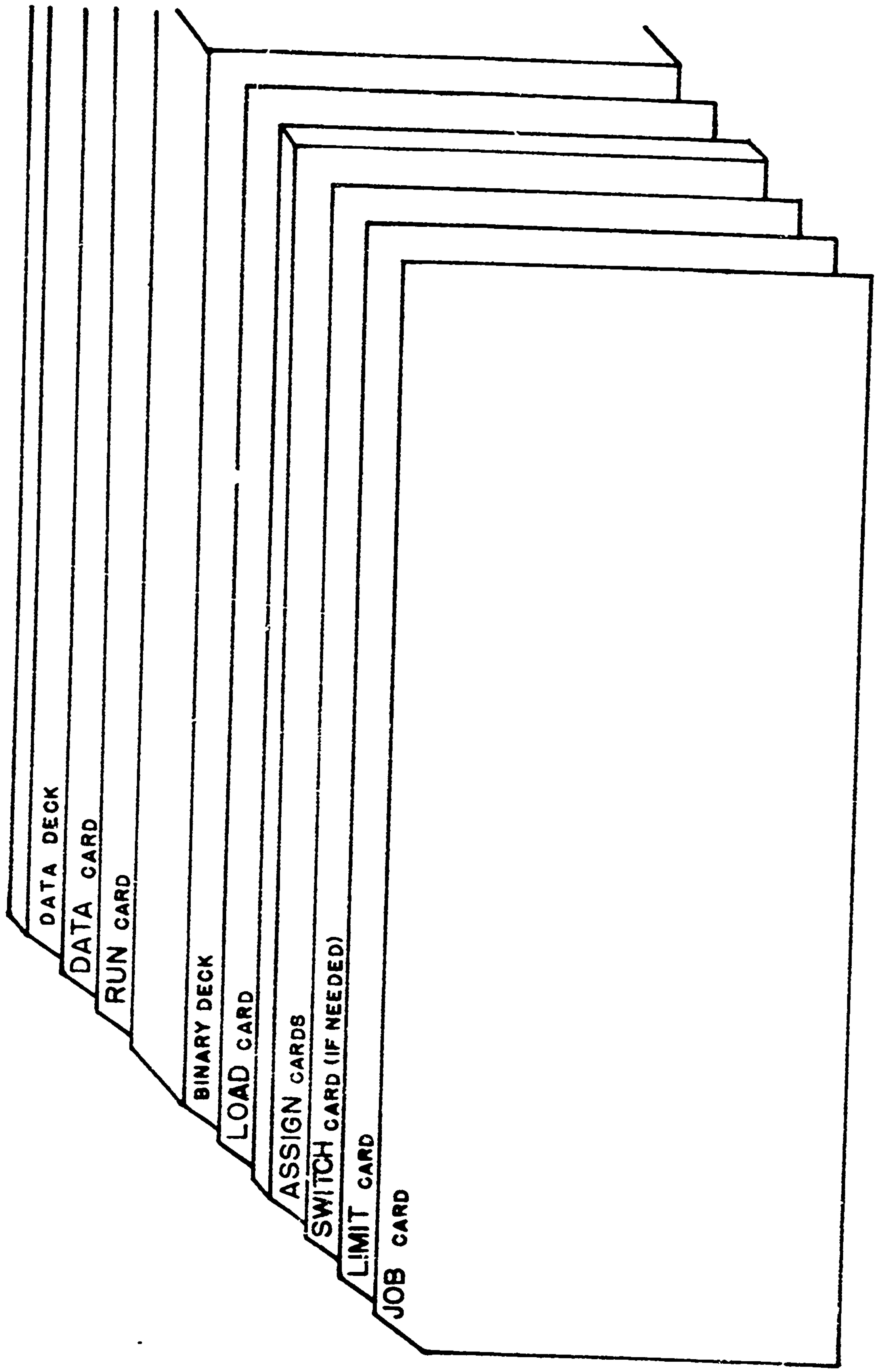


Figure 23 -- TASK 2 PROGRAM DECK

2.3.4.1 Initial Startup. The initial startup procedures is thus:

- prepare the deck as shown in Figure 23 modifying the !ASSIGN cards to indicate the proper serial numbers and modifying the cards following the !DATA card as desired. Place this deck into the card reader and then start the job in the normal manner under BPM. It is also convenient to mount the required tapes onto the tape drive without dialing unit addresses.

- When the program begins execution, the typewriter will type

- EXECUTION COMMENCING. MOUNT FIRST INPUT TAPE. MOUNT FIRST INPUT TAPE. CONTINUE WHEN READ.

- It should be pointed out that this message doesn't completely fit on the teletypewriter line and the last few characters may overprint one another. This, however, is not a serious problem. The light on the typewriter will now light up. Once the first input tape has been mounted onto a tape drive, the new line button the typewriter should be pressed. The system will now respond with the following message:

  !!MOUNTT9TA80,iii1

  where iii1 is the serial number of the first input tape. The drive which has the first input tape mounted upon it should be dialed to the zero position, the tape should be loaded (i.e., the load light should be lit) and the buttons ATTENTION,START should be pressed in that order.

- After awhile the system will type

  !!MOUNT 9TA81, ooo

where oooo is the serial number of the output tape. The procedure used in response to the similar message for the input tape should be followed for the output tapes except that the tape drive should be dialed to 1.

The program is now executing normally and will continue to do so until a new tape is required or until the program finishes.

2.3.4.2 Switching Input Tapes. The procedure for switching input tapes is as follows:

- When it becomes necessary to read the second input tape the following message will be typed on the typewriter

  MOUNT NEW INPUT TAPE. CONTINUE WHEN READY.

  Again this message will not entirely fit on the typewriter, but this is no problem. The typewriter lamp will then light. When the second input tape has been mounted onto a tape drive the new line button should be pressed on the typewriter.

- The system will now type

  !!MOUNT 9TA83,iii2

  iii2 is the serial number for the second input tape. In response to this message the tape drive which contains the second input tape should be dialed to 3, loaded, and the ATTENTION and START buttons pressed in that order.

The system is now processing with the second input tape.

2.3.4.3 Taking a Checkpoint and Continuing. The procedure for taking a checkpoint and continuing is as follows:

- If this is the first checkpoint, do nothing; if this is other than the first checkpoint, press RESET button on the drive containing the checkpoint tape.

- Press the INTERRUPT key on the console, the system will respond

  !!KEYIN

- Type "!SWITCH nnnn,(SET,1)" where nnnn is the ID of this job. ID of this job is printed in large numbers on the first sheet of paper coming from the line printer for this job.

- The system will eventually respond

  ENTER CHECKPOINT ID

  and the typewriter lamp will light. Type any three unique characters. These three characters must be different each time a checkpoint is taken. For example, within a particular run the first checkpoint ID should be AAA, the second BBB, the third CCC, and so on. CAUTION: Failure of the checkpoint ID to be unique will make it impossible to recover at this particular checkpoint.

- The system will type

  !!9TA8n MANUAL

  where lower case n is the number which has been assigned to the checkpoint tape drive. If this is the first checkpoint the system will instead type

!!MOUNT 9TA82,ccc

where lower case cccc is the serial number of the checkpoint tape. In this case the standard mount procedures as described above for the input tapes should be followed instead.

WARNING: DO NOT YET RESPOND TO THIS MESSAGE. YOU WILL BE BE INSTRUCTED TO DO SO IN A LATER STEP.

- Press INTERRUPT on the console. The system will type

  "!!KEY IN"

- Type

  !!SWITCHnnnn,(RESET 1),(SET 2)

  where nnnn is the job ID as mentioned above.

- The system may or may not give you another

  "!!9TA8n MANUAL"

  message. In any event, press START on the tape drive containing the checkpoint tape. (If this is the first checkpoint, the mount procedure should not be followed as described above for the input tapes).

The system will now continue processing.

2.3.4.4 Taking a Checkpoint and Terminating. The procedure for taking a checkpoint and terminating is the same as the procedure for taking a checkpoint and continuing except that instead of typing

!!SWITCH nnnn,(RESET,1),(SET,2)

the following should be typed:

!!SWITCH nnnn,(RESET,1,2)

The program will now come to an orderly halt. Near the bottom of the output from the lineprinter should be STOP 99999.

2.3.4.5 Restarting from a Checkpoint. The procedure to restart from a checkpoint is as follows:

- In the input deck, immediately following the !LIMIT card, place the following card:

  !SWITCH (SET,2)

  If the binary deck program is still recedent on the disc, the user file Task 2 binary decks may be removed as described above. Mount the required tapes on the drives but do not assign numbers. Then start the job in the normal BPM manner.

- The system will now type

  ENTER CHECKPOINT ID

  and the typewriter lamp will light. Enter the three characters which were typed at the time the desired checkpoint was taken. This is usually the checkpoint which was taken when the program was last terminated.

The system will now type

!!MOUNT 9TA80,cccc

where cccc is the serial number of the checkpoint tape. The unit containing the checkpoint tape is to be dialed to zero, the ATTENTION button should be pressed and then the START button should be pressed. The tape will now move forward very rapidly for a brief period of time. The more checkpoints that have been taken the longer the tape will move.

- The system will now type

  !!MOUNT 9TA81,ooooo

  where ooooo is the serial number for the output tape. The drive containing the output tape should be dialed to one, the ATTENTION button should be pressed and the START button should be pressed. This tape will now proceed to move forward for some period of time. If the checkpoint is taken near the end of the run the tape may move forward for as much as five minutes.

- The system will now type

  !!MOUNT 9TA82,iiiii

  where iiiii is the serial number of either the first or second input tape whichever was the last one being processed The drive containing the appropriate input tape should be dialed to 2, the ATTENTION button pressed, and the START button pressed. This tape will also move forward for some period of time. If the checkpoint was taken soon before the input tapes were switched the tape may move forward as long as 15 minutes.

The program will now continue processing normally.

3. PRODUCE CULTURE SWEEP PROFILES.

As stated in Section II.3 the job of Task 3 is to simulate the special-purpose hardware of the radar landmass simulator to produce sweep profiles. The purpose of this non-real time simulation is to experimentally determine those criteria -- especially in the terrain profile generation -- that will enable the government to produce satisfactory simulation hardware at the lowest possible price. Therefore, it was felt that greater flexibility could be obtained in the terrain profile generation writing it as a separate program. Task 3 consists of two programming tasks: Task 3A to produce culture profiles, Task 3B to produce terrain profiles. Task 3B is discussed in Section III.4; Task 3A is discussed here.

The software system described in Ref. 4 was taken and modified to form the Task 3A program. Section III.3.1 describes those aspects of the Task 3A program that result from modifications to the existing RLMS software. In addition to modifications to the RLMS software two programs -- CFCONT and MISCON -- were completely rewritten. CFCONT -- the program to simulate the scan converter -- is described in Section III.3.2 MISCON -- the program to simulate the motion of the aircraft -- is described in Section III.4.3 (in the section on terrain profile generation). Section III.3.3 serves as an operators manual for the Task 3A program. Details on the individual routines written and modified to produce the Task 3A program are in Ref. 10.

3.1 RLMS SYSTEM MODIFICATIONS. This section describes those aspects of the Task 3A program that result from modifications to the RLMS system software produced previously. An understanding of the system as described in Ref. 4 is necessary for understanding of this section. Changes were made in the following areas:

- the control program
- the UCB-TCB list
- input/output control tasks
- the ABEND task
- the CFPREP task.

3.1.1 The Control Program. Although the control program currently in operation would suffice for the Task 3A program, certain enhancements would increase system speed, and ease debugging. These are:

- Interrupt Tracing
- Register Set Usage
- Waiting for Itself
- Fix Discovered Bugs

3.1.1.1 Interrupt Tracing. It is very helpful when determining the cause of a system failure to know what the system was doing at the time of the failure. In straightforward programs the location of the last instruction is usually sufficient to indicate what the system was doing. In multi-programmed systems -- such as the RIMS system -- the location of the last instruction is not always sufficient. One helpful piece of information is a listing of the last few interrupts processed by the system. This will tell where in the system cycle the system was at the time of the failure and how the task which contains the failing instruction came to have control at the time of the failure.

The control program has, therefore, been modified to store the following information each time it processes an interrupt:

- The task in control at the time of interrupt
- The type of interrupt
- The PSD at the time of interrupt

3.1.1.2 Register Set Usage. RIMS Phase 2 uses one register set for the control program, and one register set for computational tasks. The machine has four register sets. By assigning a register set to a task when that task is started, and releasing it when the task completes, the control program only need save registers in core when there are no free register set available. Thus, the time to switch control from task to task is reduced.

3.1.1.3 Waiting for Itself. Empirical evidence indicates that the "waiting for itself" flag, which makes it possible to start a running computational task, does more harm than good, since it prevents RIMS programmers from debugging the system cycle. Attempting to start a running computational task is now an error condition and treated accordingly.

Starting an I/O control task that is in operation will still cause the calling task to wait for the current I/O operation to complete. No change has been made in this portion of the control program.

3.1.1.4 Fix Discovered Bugs. The control program failed to mark the I/O control task READY when it was given control in response to an interrupt. This would occasionally cause the system to permanently enter the WAIT state. This has been corrected by inserting the appropriate coding into the interrupt handling portion of the control program. (See Ref. 10 for the details.)

3.1.2 The UCB-TCB List. Table 3 lists the UCB-TCB list for the Task 3A program. There are twelve entries; of these five are UCB's and seven are TCB's.

- $TYPE - This task controls the output to the console typewriter. It is the same as the Phase 2 system.

- $PRINT - This taks controls output to the printer. It is the same as the Phase 2 system.

- $CARD - This task controls input from the card reader. It was added for the Task 3A program.

- $TAPE - This task controls input from and output to magnetic tape units. It was added for the Task 3A program.

TABLE 3

UCB-TCB LIST FOR THE TASK 3A PROGRAM

NUMBER	NAME	TYPE	ARGUMENTS
1	$TYPE	UCB	TYPESIO,TYPEINT,X'001'
2	$PRINT	UCB	PRTSIO,PRTING,X'002'
3	$CARD	UCB	CRDSIO,CRDINT,X'003'
4	$TAPE	UCB	TAPESIO,TAPEINT,X'080'
5	$RAD	UCB	RADSIO,RADINT,X'0F1'
6	$TYPESERV	TCB	TYPESERV,X'A0'
7	$ABEND	TCB	ABEND,X'A0'
8	$READREG	TCB	READREG,X'A0'
9	$MISCON	TCB	MSCN,X'00'
10	$TFCONT	TCB	TFCONT,X'A0',($CFCONT,$CFPREP)
11	$CFCONT	TCB	CFCONT,X'00',($CFPREP,$CFCONT)
12	$CFPREP	TCB	CFPREP,X'00'

- $RAD - This task controls the input from the RAD. In addition to routine I/O servicing, this task does special processing when a horizontal strip of regions is being read for the CFPREP task. It is the same as the Phase 2 system.

- $TYPESERV - This task enables the buffering of messages on the on-line typewriter. It is the same as the Phase 2 system.

- $ABEND - This task prints and error message, the contents of the current registers, the error PSD, and all of core memory, in response to the control program's detection of an error. This task was modified for the Task 3A program.

- $READREG - This task is called upon by the CFPREP task to read culture data from the RAD, a strip of regions at a time. Having this routine as a seperate task permits overlap between input/output on the RAD and computation. This task is the same as the Phase 2 system.

- $MISCON - This task is the primary task which performs the aircraft simulation. It reads cards which define the flight path to be followed. This task was rewritten for the Task 3A program. It calls upon a FORTRAN subroutine to do the actual work. This subroutine is also used by the Task 3B program and is described along with the rest of the Task 3B program in Section III.4.

- $TFCONT - This is the primary task which -- in the Phase 2 system -- controlled reading of the scan converter. It is replaced by a dummy -- do nothing -- routine for the Task 3A program. It is retained within the UCB-TCB list to maintain the system cycle. Exiting from TFCONT causes MISCON and CFCONT to be started.

- $CFCONT - This primary task simulates the action of a scan-converter by examining the culture window produced by CFPREP and generating polar sweeps. This task was written especially for the Task 3A program and is the heart of it.

- $CFPREP - This primary task brings culture data from the RAD and assembles it into a window for processing by CFCONT. As the simulated aircraft flies, this window is updated appropriately. Minor modifications were made to this task for the Task 3A program.

3.1.3 Input/Output Control Tasks. The Task 3A program must communicate with two I/O devices not used in the Phase 2 system: the card reader, and the tape drives. I/O control tasks were written for these devices. Details of these programs are in Ref. 10.

3.1.4 The ABEND Task. The Phase 2 dump routine, although useful, was found not sufficiently general. The dump program simply printed each location in hexidecimal. The improved version now also prints each location in EBCDIC and prints the mnemonic of the location's op code. The mnemonic also includes an "*" if the location specifies indirect addressing. The highest location dumped was changed to 48K to account for the additional core storage in TRADEC.

3.1.5 The CFPREP Task. Minor modifications were made to CFPREP to allow for initializing the culture window from any point in the problem area and at any time in the flight in response to a flag set my MISCON. Also a special version of the subroutine DVCT which converts the data from the RAD into internal format and setups the specularity table has been written. This special version of DVCT replaces specular objects with non-specular entries in the window. This deletes specularity as requested by NTDC. Details of these changes are described in Ref. 10.

3.2 SCAN CONVERTER SIMULATION. The Phase 2 RIMS system writes the culture data onto a scan converter in horizontal scans, and then reads the scan converter in radial sweeps and sends this data to the final display. For Task 3A program no scan converter is used. The CFCONT program, which formerly wrote the horizontal data on the scan converter and caused the data to be transmitted to the display, is replaced by a program which simulates the scan converter. This program produces a list of intensities for each sweep corresponding to the intensity profile resulting from reading the scan converter.

The TFCONT task in the Phase 2 RIMS system calculated the initial point, angle, and length of each sweep. This was then fed to a sweep generator. For the Task 3A program this sweep generator is simulated. The simulated sweep generator is used as a subroutine to CFCONT, and is discussed in Ref. 10.

Figure 24 is a main flowchart of CFCONT. The subroutine which computes the intensities is explained in the following text.

Examine Figure 25, the window being used is a 10 x 10 window. Its appearance if it were to be displayed is shown on the top. The numbers are intensities, the diagonal line is a radial scan. Below the "core" format commands to produce the top five lines of the display are shown. The commands marked with a "*" are the commands which are examined for the converted intensity. The converted intensities are 3, 4, 5, 5, and 3.

The program is called once for each radial scan line to be produced. The first call, in addition to creating the first scan line, produces entries in a table of 832 entries. Each entry corresponds to a horizontal raster line in the window. These entries are filled in by the programs the first time it needs to use an intensity code on the horizontal raster that matches the entry. Second and succeeding references to the horizontal segment proceed quite rapidly because information about where the radial sweep last passed

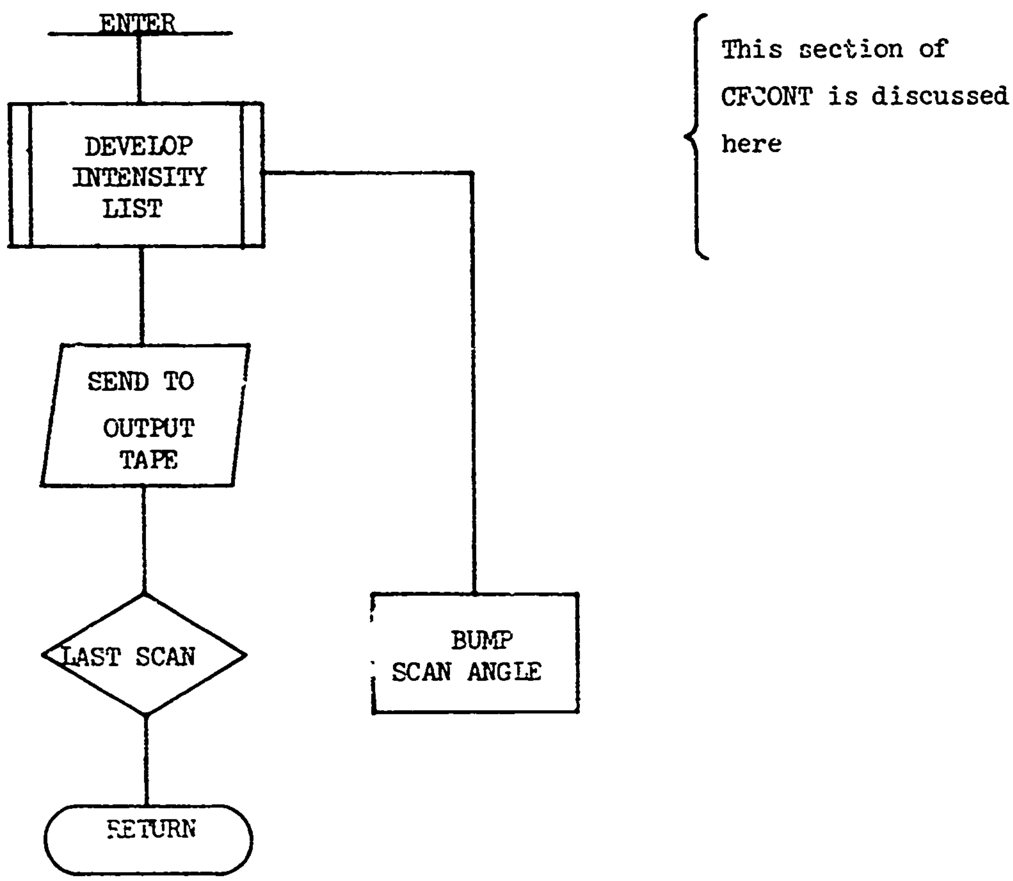
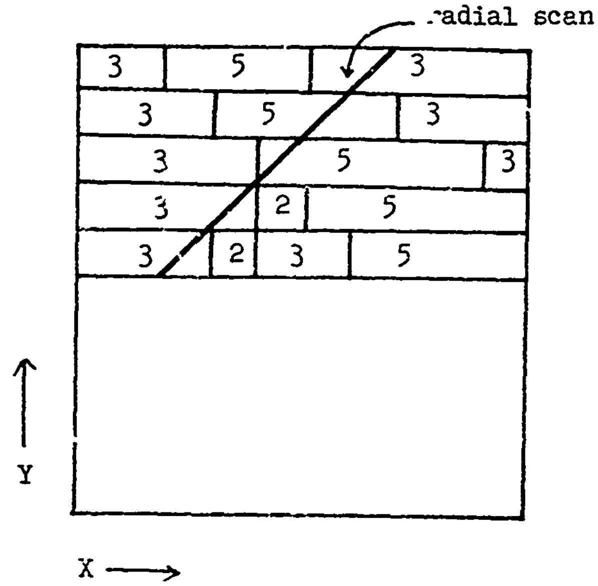


Figure 24 -- CFCONT MACROFLOWCHART



(a) IMAGE OF CULTURE
ON A SCAN CONVERTER

core location	f	ΔX	I	
x	0	3	3	*
x+1	0	1	2	
x+2	0	2	3	
x+3	1	4	5	
x+4	0	4	3	*
x+5	0	1	2	
x+6	1	5	5	
x+7	0	4	3	
x+8	0	5	5	*
x+9	1	1	3	
x+10	0	3	3	
x+11	0	4	5	*
x+12	1	3	3	
x+13	0	2	3	
x+14	0	3	5	
x+15	1	5	3	*

(b) THE SAME IMAGE IN RUN-LENGTH CODE
("CORE" FORMAT)

Figure 25 -- CULTURE DATA FORMATS

is used as a first order approximation of where it will go next. Since the variation in angle between a given sweep and the next sweep is small, the next sweep strikes very close to the previous sweep. Knowing where that was speeds processing. The table consists of entries which include both the location of the command last referenced on this line, and the sum of of the delta x's on this line which preceded the delta x of the command pointed to.

Table 4 shows the table produced after the radial scan line in Figure is produced. The location of the commands used to form the radial scan and the sum of delta x's which preceded the segment pointed to on the scan line.

The CFCONT -- Develop Intesity List flowchart (Figure 26) shows the method of stripping off the desired intensities and storing them, using the table of pointers described above. Numbers in parenthesis indicate flowchart block numbers.

The first two blocks are shown to indicate that they are needed, but should not be located in the program when it is implemented, since they will prevent maximum efficiency. The boxes in question, first check to see if there is room to create the table. (1) If not, the program takes an error exit, indicating no core available. Next the pointer table is cleared (2).

This process is normally done only once for each horizontal to radial conversion, while this program would clear the table each time it is called. (The program is called once for each radial scan line.)

TABLE 4

HORIZONTAL RASTER SEGMENT POINTERS

Horizontal Scan	Core Location	$\Sigma (\Delta X)$
1	0	-
2	0	-
3	0	-
4	0	-
5	0	-
6	x	0
7	x+4	0
8	x+8	4
9	x+11	3
10	x+15	5

A subroutine is called to get the X and Y coordinates of the intensity desired on the window. (3) If the routine indicates that a complete radial scan has been converted, the Develop Intensity List program returns. (4) The Y coordinate returned by the get X,Y routine is used to index the pointer table. (5) If the entry in question is zero, (6) the program must manually skip to the horizontal raster in question, using the "f" bit of the commands to count horizontal raster segments. (7) When the segment in question is found, the sum of delta x's is set to zero (8), and the pointer is set to show the first command on the line (9). If the pointer was not zero, the sum of delta x's is moved to SDX. (9.1)

Whether or not the pointer was zero, the program is now ready to extract the intensity from the line. The sum of delta x's is compared with the X coordinate needed. (10) If the sum is less than X, the program must travel down the line segment. The "next" command is fetched, (this is the command pointed to by the pointer table). (11) SDX is incremented by the value of the delta x in the fetched command. (12) The incremented value is compared with X, to see if we have reached a command which includes the point we are interested in. (13) If not, the program loops back to pickup the next command. (11) If so, the program is ready to collect the required intensity.

If the sum of delta x's was greater than the X coordinate required, the command currently pointed to is saved. (17) The previous command is fetched (18), and its delta x is decremented from SDX. (19) SDX is then checked to see if it is now less than X. (20) If it is not, we back up another command in check again (17). If SDX is now less than X, the previous command is the command which has the intensity required. This command is picked up out of the save area. (21)

The intesity required has been found, and is handed to the store program. (14) The pointer to the command most recently used in this raster segment is stored back in the pointer table (16), and the program returns to the beginning (3) to perform another major cycle.

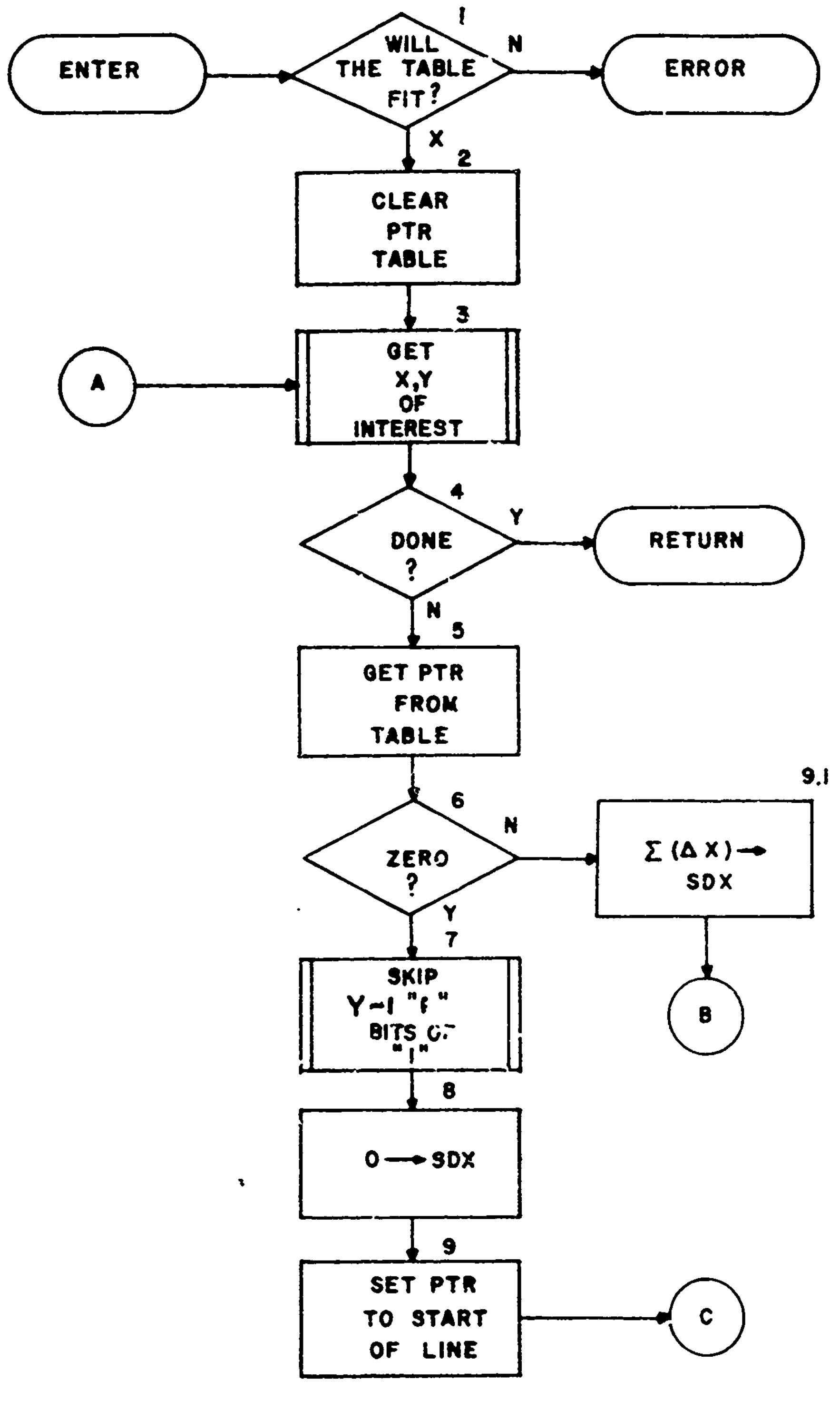


Figure 26 -- CFCONT - DEVELOP INTENSITY LIST
(Sheet 1 of 2)

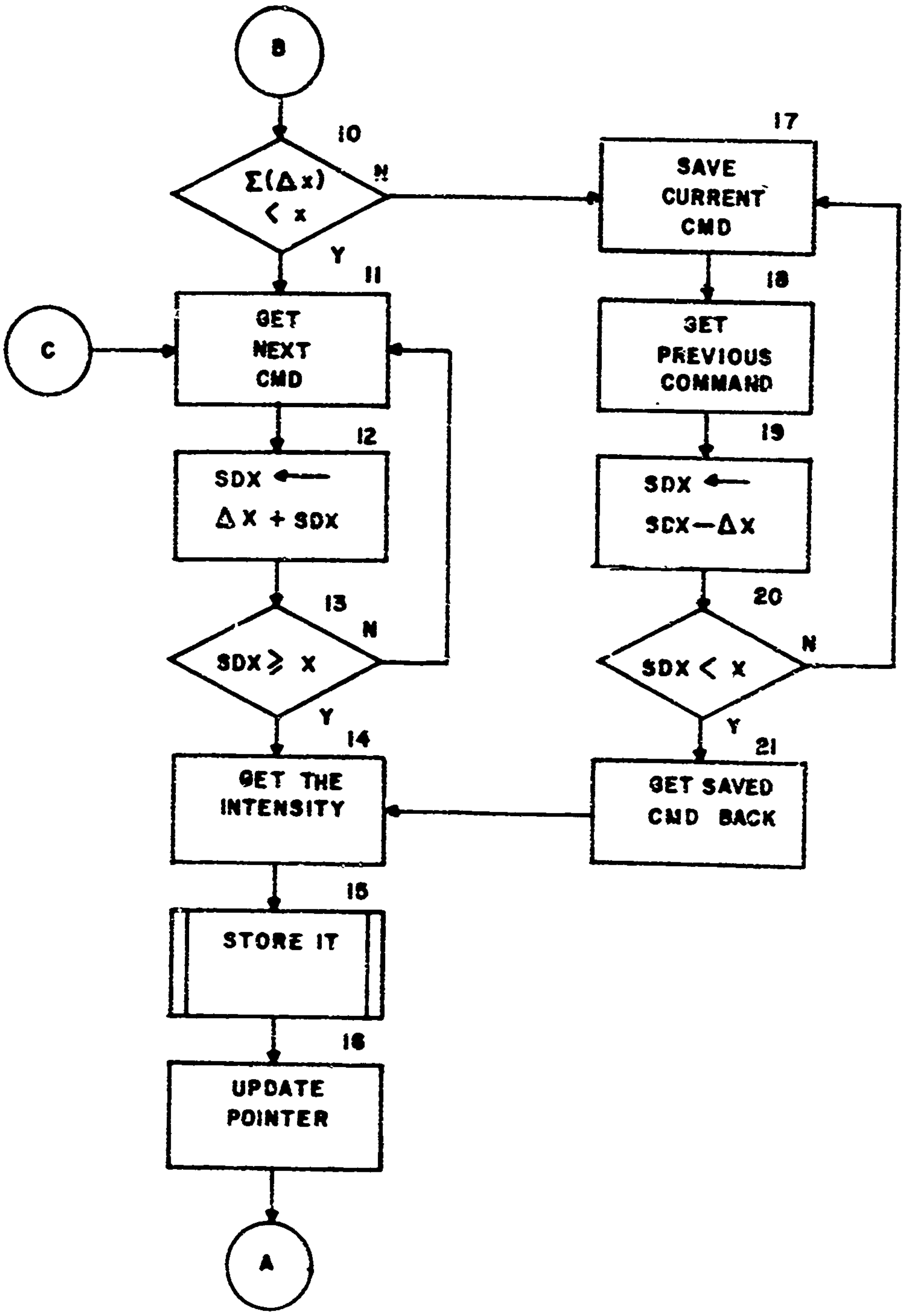


Figure 26 -- CFCCNT -DEVELOP INTENSITY LIST
(Sheet 2 of 2)

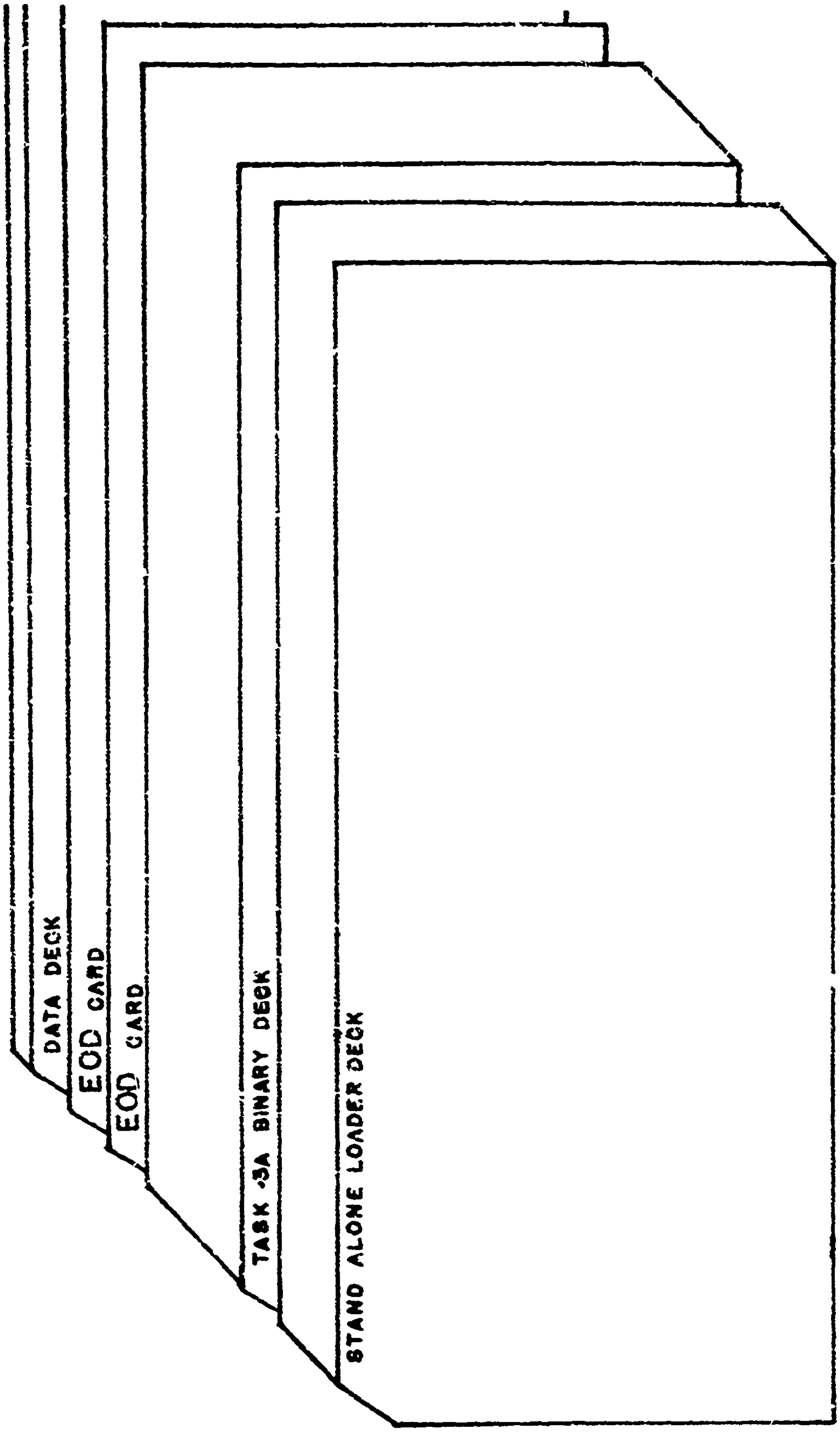


Figure 21 -- TASK 3A PROGRAM DECK

3.3 PROGRAM OPERATION. This section serves as the user's manual for Task 3A program. Figure 27 shows the deck setup. The preparation of data cards is given in Section III.4.4.1. The data cards used should be the same ones used to prepare the terrain profile tape with the Task 3B program. This will guarantee that the two tapes correspond.

To operate the Task 3A program:

- Verify that the RIMS data base has been loaded onto RAD OF1. If not load it.
- Prepare the deck into the card reader.
- Mount the output tape onto 9TA80
- Use the standard load sequence to load the standalone loader. The typewriter will indicate it is ready for input.
- Type "!L RB000,MAP,N" into the typewriter, followed by the n/l (new line) key. The RIMS binary decks will the be read, and the printer will produce a memory map. The typewriter will again indicate it is ready for input.
- Type "!R BEGIN". The RIMS system will begin executing, after n/l key is pressed.

4. PRODUCE TERRAIN PROFILES. It was felt by NTDC that greater flexibility would be achieved if the program to produce terrain profiles were written in FORTRAN as a separate program rather than as in assembly language (METASYMBOL) as a task under the RIMS system. Section IV.2 discusses the consequences of this decision as it relates to the resulting program efficiency. This section describes Task 3B to a macro level of detail. Ref. 11 gives micro details of the individual program.

4.1 OVERALL DESCRIPTION OF THE PROGRAM. Figure 28 is a structural diagram of the Task 3B program; Figure 29 is a macroflowchart of this program. The MISCON subroutine is the interface between the flight control deck and the program. The TFPREP subroutine (along with its subroutines) is the interface between the data base, which is stored onto the RAD, and the program. The TFCONT subroutine (along with its subroutines) is the simulator of the special-purpose hardware to be used in the real-time system.

The terrain portion of the system cycle is simulated. There are three steps:

(1) Obtain the new location of the aircraft
(2) Update the window
(3) Generate a scan

Step (1) is performed by the MISCON subroutine; step (2) is performed by the TFPREP subroutine; and step (3) is performed by the TFCONT subroutine. Macro descriptions of the MISCON and TFCONT subroutines are given in the next two subsections.

4.2 MULTRIX SIMULATION. Figure 24 serves equally well as a TFCONT macroflowchart except that the subroutine to compute intensities is replaced by a subroutine which computes heights. This subroutine simulates the multiplying matrix (multrix) which will evaluate the height approximation polynomial in real time in the Phase 3B simulator.

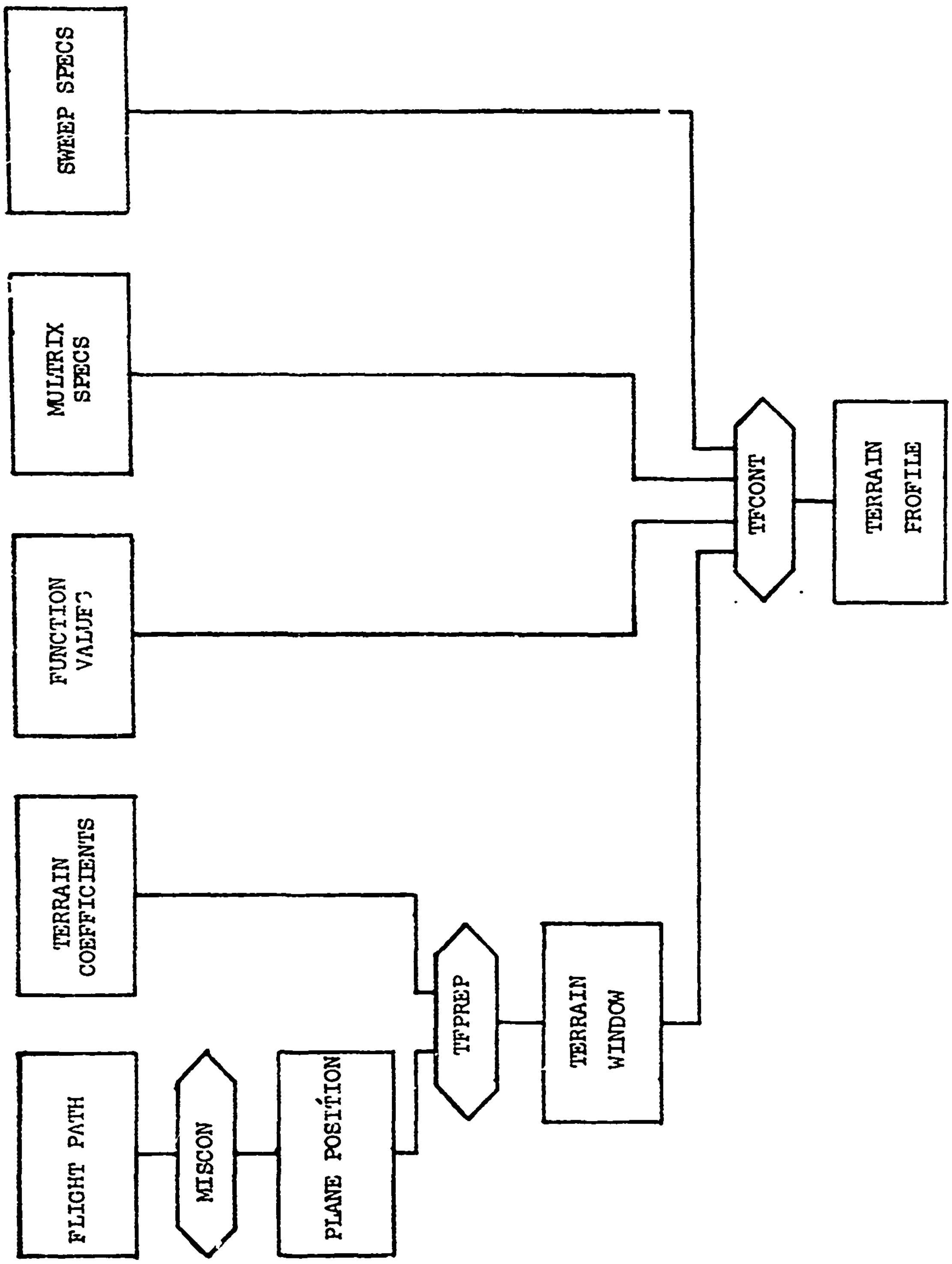


Figure 28 -- TASK 3B PROGRAM STRUCTURE

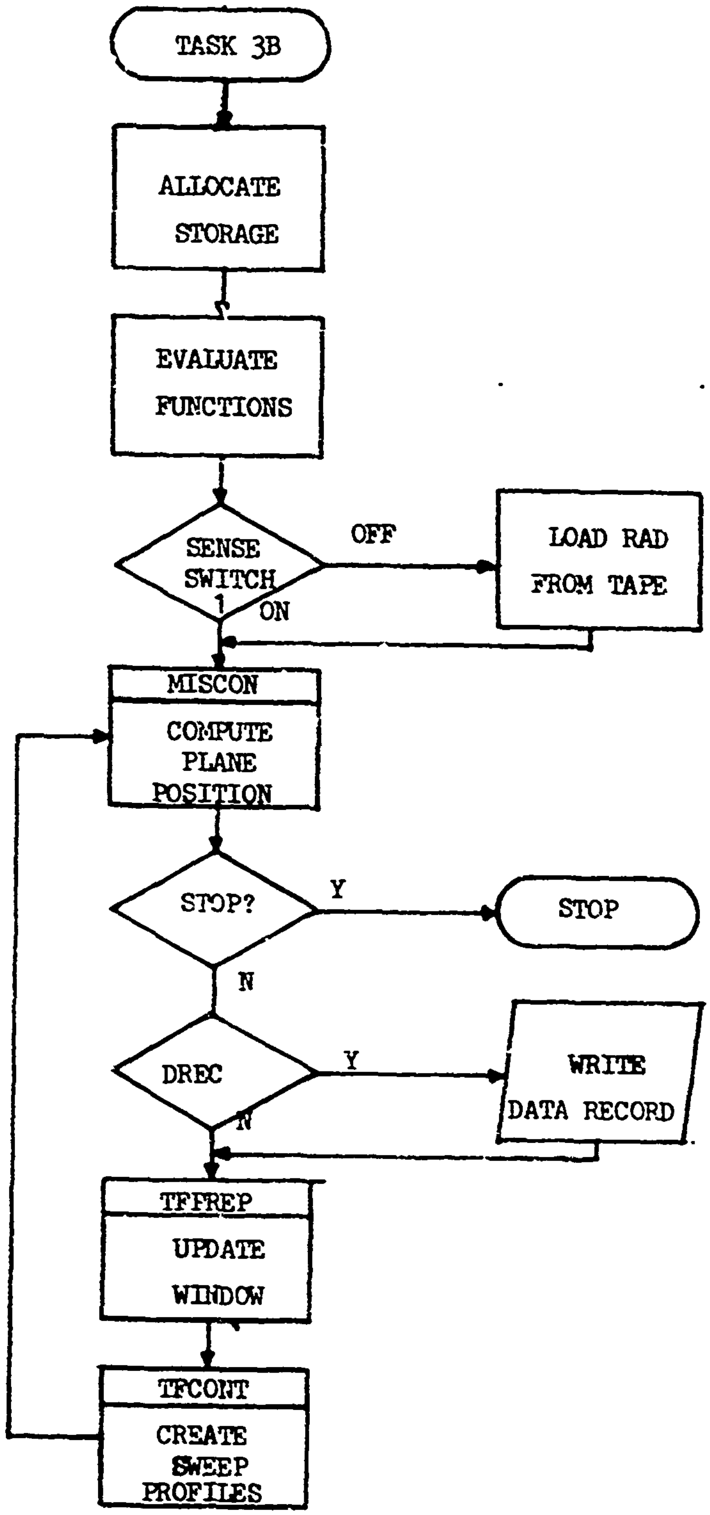


Figure 29 -- TASK 3B MAIN PROGRAM MACROFLOWCHART

The process of finding which regions are of interest which was discussed in Engineering Report 6 (Ref. 6) is much simpler for the Phase 3A system. Since each resolution element of a sweep is processed, the high order bits determine which coefficients are fed to the simulated multrix and the low order bits are the X and Y fed to the simulated multrix.

The multiplying matrix (multrix) performs the fixed-point calculation for each resolvable point in the region:

$$z(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{p=0}^{n} \sum_{q=0}^{n} c_{ijpq}\, g_{ip}(x)\, g_{jq}(y) \qquad (1)$$

where $z(x,y)$ = reconstructed terrain height at point (x,y)

c = coefficients for the region

g = basis functions (e.g., Lagrange polynomials)

i, j = corner of the region from which c's are drawn

p, q = order of the basis function

n = degree of representation used.

The actual calculation to be made is:

$$Z(x,y) = \sum_{i=0}^{n} \sum_{j=0}^{n} \sum_{p=0}^{n} \sum_{q=0}^{n} \left[C_{ijpq}\, G_{ip}(x)\, G_{jq}(y) \right] \qquad (2)$$

The bracketed term is developed by a replicated set of hardware, which is then sequenced among the four corners. The capital letters indicate normalization:

$$G_i = N_i\, g_i \qquad\qquad G_j = N_j\, g_j$$

$$C_{ij} = \frac{c_{ij}}{N_i\, N_j}$$

The N_i, N_j are chosen so as to confine the G_i, G_j to the range $-1 < G < +1$; the values of N depend on the form of the G .

Establish the convention that σ represents the exponent of the most significant digit and τ represents the exponent of the least significant digit in a fixed-point number.

A twos-complement binary number having $\sigma = +1$, $\tau = -3$ for example, would contain five bits and look like:

2^{+1}	2^{0}	2^{-1}	2^{-2}	2^{-3}

Δ
radix
point

In like manner a positive decimal number having $\sigma = +2$, $\tau = +1$ would have two digits and look like:

10^{2}	10^{1}	

Δ
radix
point

The σ, τ method of indicating precision and radix point location is used for simulating the multrix; σ, τ values will be provided as input parameters at strategic points in the simulated calculation. The operation "delimit" is defined: truncate the most significant digit to the σ value and round the least significant digit at the τ value. This delimit operation is implemented as a METASYMBOL procedure so as to produce a more efficient program.

The multrix is anticipated to operate in binary, with twos-complement arithmetic being employed.

Three σ, τ pairs are needed to specify the multrix's implementation of eq. (2). The basis functions G, the coefficients C, and the partial sums (in brackets) all have limited register sizes. Multrix simulation can be formulated as shown in Figure 30 for any point (x,y) in the region. Obviously the calculation of Figure 30 can be iterated over all points comprising a profile through any number of regions.

4.3 DESCRIPTION OF MISCON PROGRAM. The program to simulation the motion of the aircraft (which contains the simulated radar antenna) has been rewritten to accept a preprogrammed flight path.

The experimenter initializes the problem by specifying an initial position.

- distance east of origin
- distance north of origin
- altitude

and an initial velocity vector:

- course
- speed
- rate of climb
- duration of this velocity

Thereafter the experimenter specifies up to three types of acceleration by providing entries to the system containing:

- turn rate (change in course)
- longitudinal acceleration (change in speed)
- vertical acceleration (change in rate of climb)
- duration of this acceleration

Accelerations are signed numbers. A zero acceleration of any type means to retain the current velocity constant for the duration specified. Two or all three types of acceleration may be changed for the same duration. Obviously the sum of the durations is the total duration of the flight.

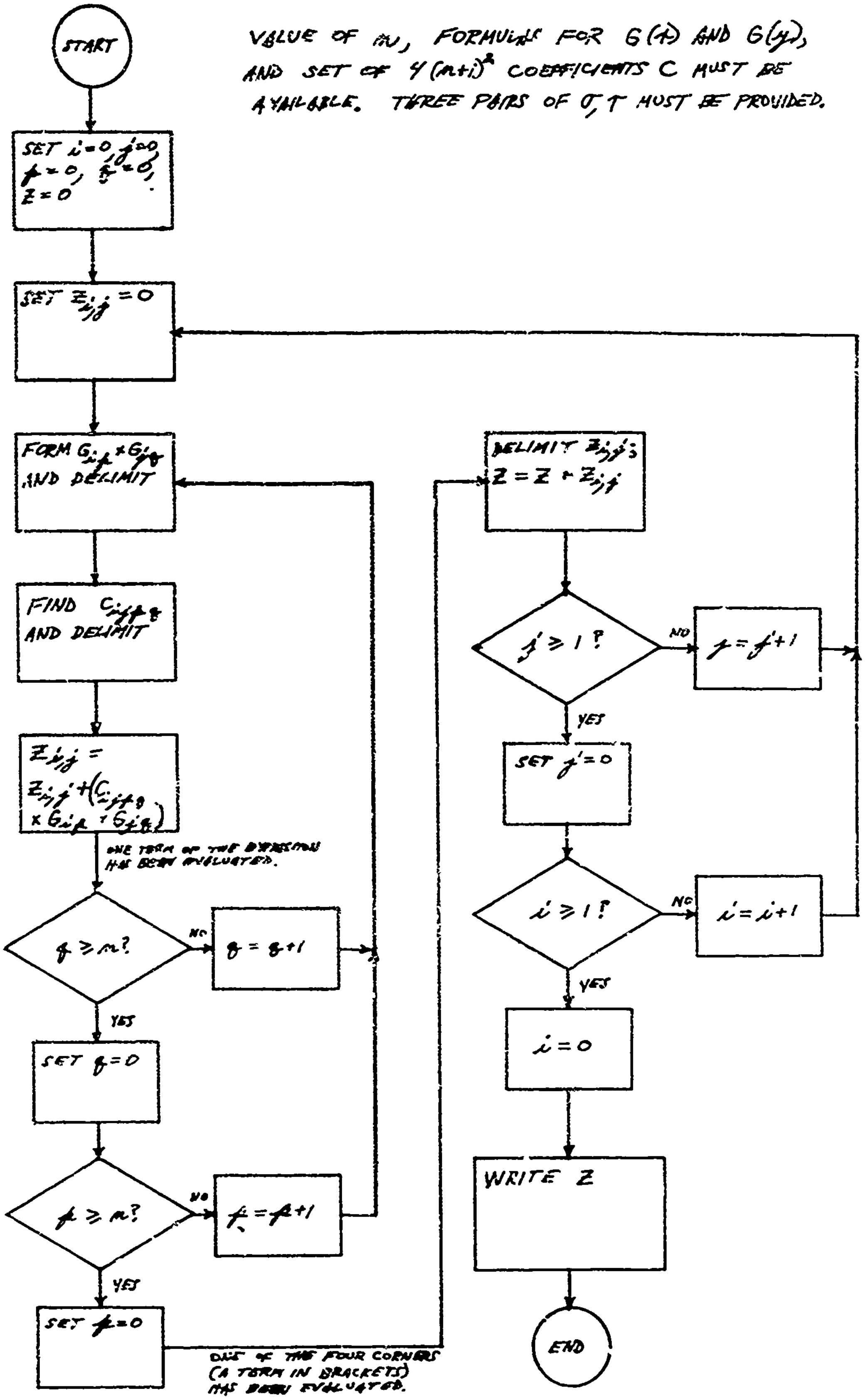


Figure 30 -- MULTRIX SIMULATION MACROFLOWCHART

The foregoing method is obvious to program yet allows convenient prediction of the result of each new entry as it affects vehicle position and orientation. The computer response is to repeat the initial position and velocity at time zero, then to give position and velocity in tabular form at specified intervals (such as each radar scan). Further, the flight path can be reconstructed from the output alone by having the tabular printout of position and velocity vs. time interspersed with a repeat of the commands.

There are three possible units in which the above quantities may be given: internal units, conventional units, and map units. Internal units express distance in regions and dots, and express time in system cycles. Conventional units are miles, feet, and seconds. Map units express position in terms of latitude and longitude. This program provides for internal and conventional units. The next section describes the details of the data cards.

4.4 PROGRAM OPERATION. This section is the user's manual for the Task 3B program. Figure 31 shows the deck arrangement. Actual computer operation is straightforward and is discussed briefly in Section III.4.4.3 below. Section III.4.4.2 discusses the other parameters that may be varied. The preparation of the data cards is not as simple. This is discussed here.

4.4.1 Preparation of the Data Cards. The data cards are used to define the flight path of the aircraft and provide miscellaneous control information. All cards are read by the same FORTRAN FORMAT. Figure 32 shows a generalized data card. There are four alphanumeric fields and four numeric fields. There are seven types of data cards. Not all fields are used for each type of data card. The individual cards are now described.

4.4.1.1 Initial Position Card. This card defines the initial position of the aircraft. The alphanumeric fields (columns 1-13) contain "POSITION" with the "P" in column 1. The first three numeric fields contain integers all right justified in their respective fields as follows:

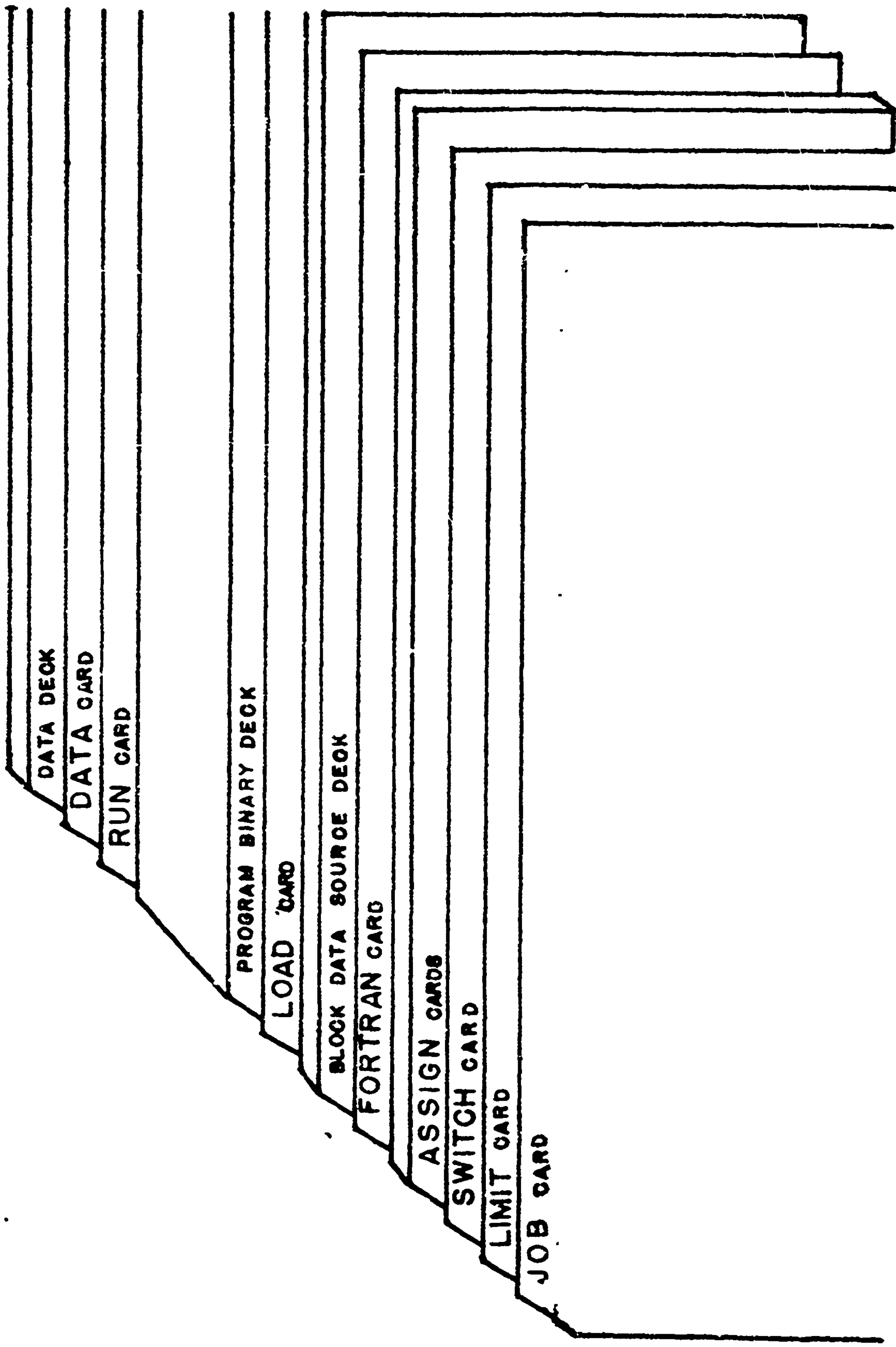


Figure 31 -- TASK 3B PROGRAM DECK

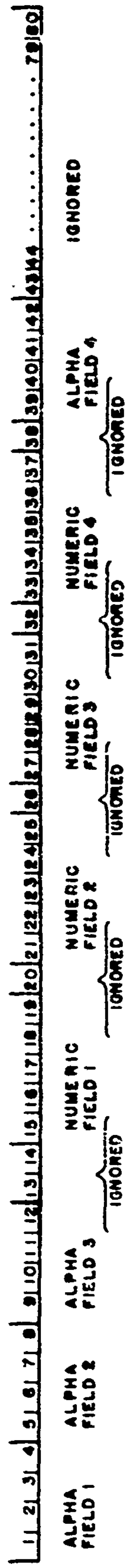


Figure 32 -- FLIGHT SPECIFICATION CARD FORMAT

Numeric Field	Columns	Contents
1	15-19	The distance east of the origin
2	21-25	The distance north of the origin
3	27-31	The altitude

In addition to these columns 39-42 indicate which set of units were used. Absence of any specification will indicate internal units (i.e., dots). If conventional units are used (i.e., feet), the letters "CONV" must be punched into columns 39-42.

4.4.1.2 Initial Velocity Card. This card defines the initial velocity of the aircraft. The alphanumeric fields (columns 1-13) must contain "VELOCITY" with the "V" in column 1. The four numeric fields all contain integers right justified in their respective fields as follows:

Numeric Field	Columns	Contents
1	15-19	The aircraft heading expressed in degrees. Heading is measured clockwise with respect to the left edge of the problem area. (At the left edge this is north. East of this edge zero becomes slightly east of north.)
2	21-25	The speed of the aircraft.
3	27-35	The rate of climb of the aircraft.
4	33-37	The duration of this velocity, i.e., the time until the next card is to be read.

In addition columns 39-42 indicate the set of units. Internal units express both speed and climb rate in dots per cycle. Conventional units express these

in feet per second. The duration is conventionally measured in seconds and internally measured in cycles. Columns 39-42 must contain "CONV" to indicate that conventional units were used.

4.4.1.3 Acceleration Card. This card indicates accelerations. The alphanumeric fields (columns 1-13) must contain "ACCELERATIONS" with the "A" in column 1. The four numeric fields all contain integers right justified in their respective fields as follows:

Numeric Field	Columns	Contents
1	15-19	The turn rate. This is expressed as the number of minutes for a full 360° turn.
2	21-25	The change in speed
3	27-31	The change in climb rate
4	33-37	The duration, i.e., the time until the next card is read.

Columns 39-42 indicate the units. Change in speed and change in climb rate are expressed conventionally as feet per second per second; they are expressed internally as dots per cycle per cycle. Duration is expressed internally as cycles and conventionally as seconds.

4.4.1.4 New Flight Card. This card signals the beginning of a flight. There may be several flights per run of the Task 3 programs. The first alphanumeric field (columns 1-4) must contain "NEWF". The second alphanumeric field (columns 5-8) contain the flight number. This same four-characters identifier must be given to the Task 4 and Task 5 programs to process this flight. It is the identifying characteristic that distinguishes one flight from another. The first three numeric fields contain integers right justified as follows:

Numeric Field	Columns	Contents
1	15-19	The beam spacing in 4096'ths of a radian.
2	21-25	The number of sweeps in a scan
3	27-31	The beam width (used only by Task 3A) in 4096'ths of a radian.

4.4.1.5 Initiate Data Recording. This card signals the MISCON subroutine should print the position of the aircraft once each cycle. The first alpha-numeric field (columns 1-4) must contain "DREC". All other fields are ignored.

4.4.1.6 Stop Data Recording. This card signals the MISCON subroutine should no longer print the data record. The first alphanumeric field (columns 1-4) should contain "NDRC". All other fields are ignored.

4.4.1.7 Stop Simulation. This card signals that the Task 3A or Task 3B program is to come to a halt. The first alphanumeric field (columns 1-4) must contain "STOP". All other fields are ignored.

4.4.2 Simulation Parameters. The Task 3B program can recontruct terrain from any data generated by the Task 2 program. The basis functions are defined by including the Task 3B program the same subroutine GG as was used in the Task 2 program. Other parameters may be set by modifying the BLOCK DATA subroutine in the Task 3B program.

The possible parameters for modification are as follows:

VARIABLE NAME	MEANING AND RESTRICTIONS
REGSIZ	The number of dots per side of a region
WSIZ	The number of regions per side in the window
K	The number of words stored per region in the window. This is equal to $n^2/2$ where n is the number of basis functions.
NXREG, NYREG	The size of the problem area in regions
IDOT	The size of a dot in 1/16 of a foot.
TIME	The time per cycle in seconds.
MAXRANGE	The sweep length in dots

The following are sigma-tau values used in the MULTRIX simulation (see Section III.4.2)

CIS,CIT	The initital sigma and tau of the coefficients
CDS,CDT	The desired sigma and tau of the coefficients
GIS,GIT	The sigma and tau for the tabulated functions
GGS,GGT	The sigma and tau for the product G(x) G(y)
CGS,CGT	The sigma and tau for the product C and G(x) G(y)
SGS,SGT	The sigma and tau for the final sum

4.4.3 Computer Operation. This program operates as a batch program under BPM. A special operating system which has over 507,455 words of FORTRAN random access disc space available must be used.

The program optionally loads the terrain data base onto the disc each time it is run. It is advisable to keep a special copy of the system taken after the Task 3B program has been run which includes the data. The "!SWITCH (SET.1)" card is used to bypass the writing of the tape onto disc.

The !ASSIGN cards are used as follows:

F:1	The disc
F:2	The data base on tape from Task 2
F:3	The output tape

5. PRODUCE INTENSITY WAVEFORM

Task 4 produces intensity profiles corresponding to sweep lines on the PPI, in compressed (run-length coded) form for later playback to the special-purpose display equipment in simulated real time (Task 5). It requires as input either or both the culture (reflectance) profiles from Task 3A and the terrain (height) profiles from Task 3B, which are combined with various "radar effects" inserted. Task 4 also performs the coordinate transformations between the culture and terrain profiles (produced in true geographic orientation and ground range) and the display coordinates (required in offset-center, sector scan format and slant range).

The insertion of radar effects such as earth curvature, target shadows, and side lobe phenomena is provided in a modular manner, so that experimentation can be performed by changing only small portions of the Task 4 program. In addition, the algorithms pertaining to display format are generalized allowing ease of expansion to displays other than the basic 45-degree sector, 20-mile range format which is being implemented at present.

The operations of this task are directed by punched cards for each flight to be processed. Sense switches are used to indicate when the last culture or terrain tapes are mounted as well as to signal an operator requested termination. Figure 33 shows a model of Task 4.

5.1 PROCEDURAL DESCRIPTION OF TASK 4 PROCESSOR. The description below may be followed in diagrammatic form as it is keyed (by number) to the macro diagram of the Task 4 processor, Figure 39, included in this report.

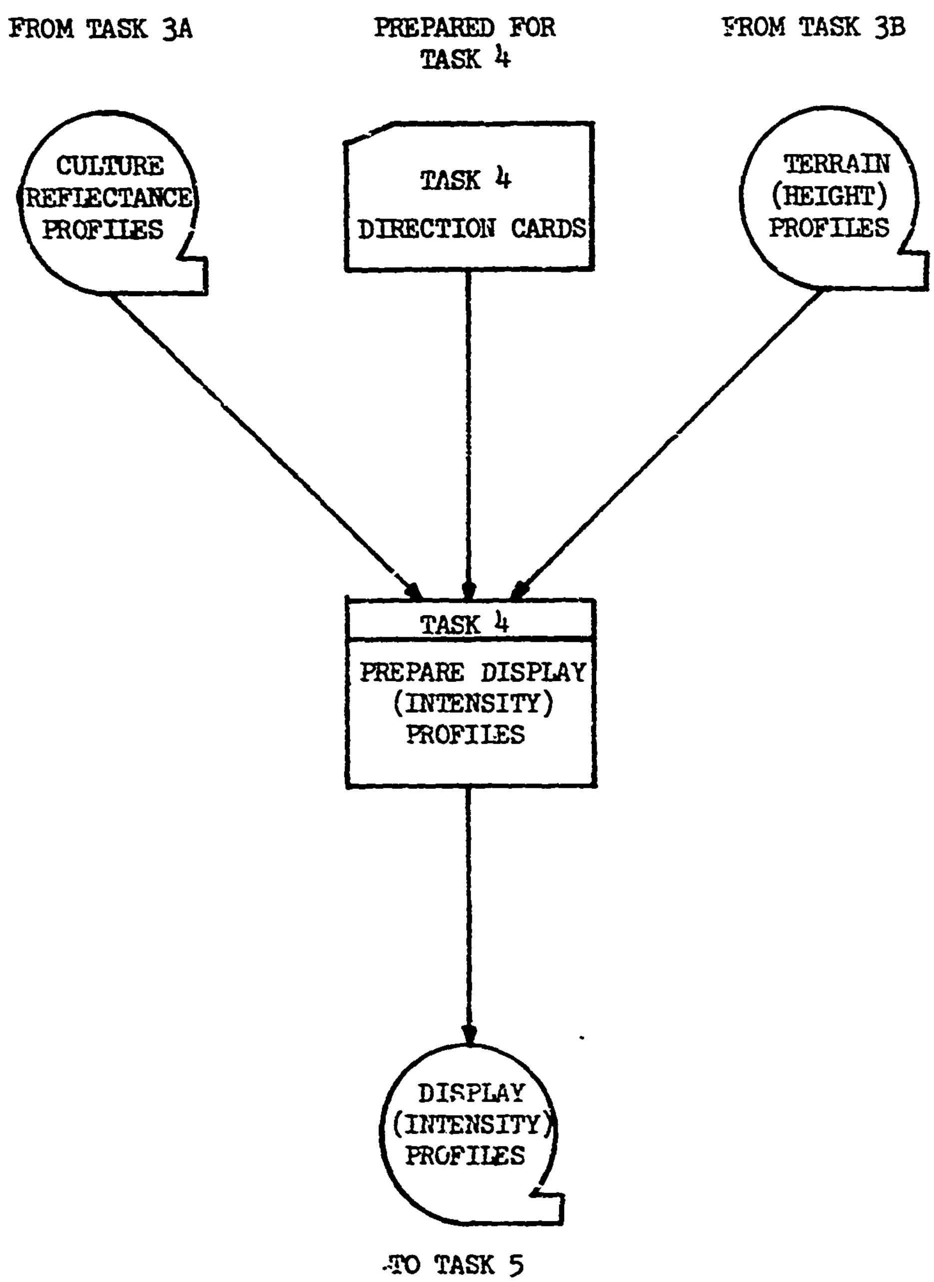


Figure 33 -- TASK 4 MODEL

Task 4 receives its processing directions via punched card input. Certain optional directions modify flight processing. These modifying options include STATUS reports to be printed BY FLIGHT or BY SCAN, SMOOTH options to smooth terrain data by a factor N, GRATICULE options to adjust the simulated graticule from dark to bright (0 to 7), and GROUND RANGE to prepare data in ground range rather than slant range. Modifying option cards are given prior to flight processing directions and remain at their selected state for all flights unless changed prior to a flight by another modifying option. If no options are given, defaults of STATUS BY SCAN, SMOOTH level 0(off) and GRATICULE light of -1 (off) are assumed. Slant range conversion is always operating unless ground range is specified for particular flight to observe ground range. Figure 34 gives a descrption and a list of allowable modifying options to flight processing.

Direction cards for flight processing should follow their modifying options (if any) with the flight numbers in the <u>same order as they appear on the input tape(s)</u>. The flight processing directions include the 4-character flight number and an option to process TERRAIN ONLY, CULTURE ONLY or BOTH FACTORS. If all flights on the input tape(s) are to be processed the same, one card specifying ALL flights for that processing option may be given. If no flight processing cards are present the program assumes ALL flights to be processed with BOTH FACTORS present. (1,2)

The description and format of Task 4 Direction cards are given in Figure 34 (DI-1). Every flight processing card must have a corresponding flight on the input tape(s). However, not all flights on the tape(s) need have a processing card. After the first directions are read and stored (3), or the default directions stored (4), the beginning of the data to be processed is looked for on the input tape(s) to verify its presence. (5) If the flight on the processing card cannot be found on the input tape (culture and/or terrain), no other flights of that type of data will be processed as tapes are only read once.

PUNCHED CARD DIRECTIONS FOR SLANT RANGE CONVERSION:

LEGAL 1st WORD		LEGAL 2nd WORD	
GROUND RANGE	DENOTES GROUND RANGE RATHER THAN SLANT RANGE CONVERSION FOR GIVEN FLIGHT.	(NONE)	SLANT RANGE CONVERSION MAY BE "TURNED OFF" TO VIEW IN GROUND RANGE.

PUNCHED CARD DIRECTIONS FOR GRATICULE LIGHT SETTING:

LEGAL 1st WORD		LEGAL 2nd WORD	
GRATICULE	DENOTES SETTINGS FOR GRATICULE LAND. IF NO GRATICULE DIRECTIONS ARE GIVEN IT IS SHUT OFF.	INTEGER 0 TO 7	(0 IS DARKEST, 7 IS LIGHTEST). -1 (GRATICULE IS OFF).

PUNCHED CARD DIRECTIONS FOR SMOOTHNESS EFFECT ON DATE:

LEGAL 1st WORD		LEGAL 2nd WORD	
SMOOTH	DENOTES LEVEL OF SMOOTHNESS. IF NONE GIVEN, SMOOTHNESS IS SHUT OFF.	INTEGER 4 TO N	(<4 IS OFF, OTHERWISE THE HIGHER THE HIGHER THE NO. THE GREATER THE SMOOTHING. N>30 IS POSSIBLE BUT NOT REASONABLE.

PUNCHED CARD DIRECTIONS FOR FLIGHT LOG (STATUS REPORTS): IF NO STATUS DIRECTIONS ARE GIVEN THE OPTION BY SCAN IS ASSUMED.

LEGAL 1st WORD		LEGAL 2nd WORDS	
STATUS	DENOTES REQUEST FOR FLIGHT LOG.	BY FLIGHT	WRITE LOG FOR EACH FLIGHT PROCESSED.
		BY SCAN	WRITE LOG FOR EVERY SCAN PROCESSED.

PUNCHED CARD DIRECTIONS FOR FLIGHT PROCESSING: IF NO FLIGHT PROCESSING DIRECTIONS ARE GIVEN THE OPTIONS ALL AND BOTH ARE ASSUMED.

LEGAL 1st WORDS		LEGAL 2nd WORDS	
NNNN	DENOTES REQUEST TO PROCESS DATA FOR FLIGHT NO NNNN.	TERRAIN ONLY	DENOTES PROCESSING OF TERRAIN DATA ONLY FOR THIS FLIGHT.
		CULTURE ONLY	DENOTES PROCESSING OF CULTURE DATA ONLY FOR THIS FLIGHT.
ALL	DENOTES REQUEST TO PROCESS ALL FLIGHTS IN ORDER OF INPUT TAPE.	BOTH FACTORS	DENOTES PROCESSING OF BOTH TERRAIN AND CULTURAL DATA COMBINED FOR THIS FLIGHT.

NOTE: ALL WORDS MUST BE LEFT JUSTIFIED IN THEIR FIELD: FIRST WORD BEGINS IN COLUMN 1, SECOND WORD BEGINS IN COLUMN 13. ONLY ONE DIRECTION CARD FOR EACH FLIGHT NO. IS PERMITTED. DIRECTION CARDS MUST BE IN THE SAME ORDER AS FLIGHT NUMBERS ON THE INPUT TAPES.

Figure 34 -- TASK 4 DIRECTION CARDS

A flight log (or status report) is written on the line printer giving specific information about the data to be processed and the option for processing requested (5A). A sample of the flight log is given in Figure 35 Flight Log Sample (DI-2). Any messages pertaining to specifics of processing that are of importance are listed under the entry for the flight (or scan). A new flight will start on a new page unless only information BY FLIGHT was requested.

If, at any time during the processing, the operator wishes to stop the job and have a complete Display Profile tape up to the presently processing scan, Sense Switch #1 may be turned on. The Sense Switch will be interrogated and the job stopped. Once the processing has been stopped, there is no recovery from that condition (5B).

If the directions for processing the flight call for CULTURE ONLY or BOTH FACTORS (6), then one sweep of the Cultural Profile tape from Task 3A is read and stored for processing by the read command (7). Figure 36 shows the format for the input tape(s) form Task 3 (DI-3). If the option for TERRAIN ONLY was requested for this flight, default values for culture would be stored (8). The default value for culture is a uniform intensity of 3 for all dots in the sweep.

The sweep data given as culture is processed by two subroutines which perform the following:

- ENHANC Performs overall enhancements such as far-shore brightening or shadowing by cultural objects, to augment the specular effects handled individually in the culture map, and inserts glint or other extraneous signals to be simulated. (9) This program operates by altering the value of each dot (resolution element) along the reflectance profile obtained as

FLIGHT NO.	VEHICLE X-COORDINATE (DOTS)	VEHICLE Y-COORDINATE (DOTS)	ALTITUDE (FEET)	HEADING (DEGREES)	STATUS (INPUT)	SCAN NO.
12K4	192	470	1200	127	BOTH FACTORS	1
12K4	217	425	1200	130	BOTH FACTORS	2
12K4	320	410	1200	135	BOTH FACTORS	3

Figure 35 -- DI-2 FLIGHT LOG (STATUS REPORT) SAMPLE

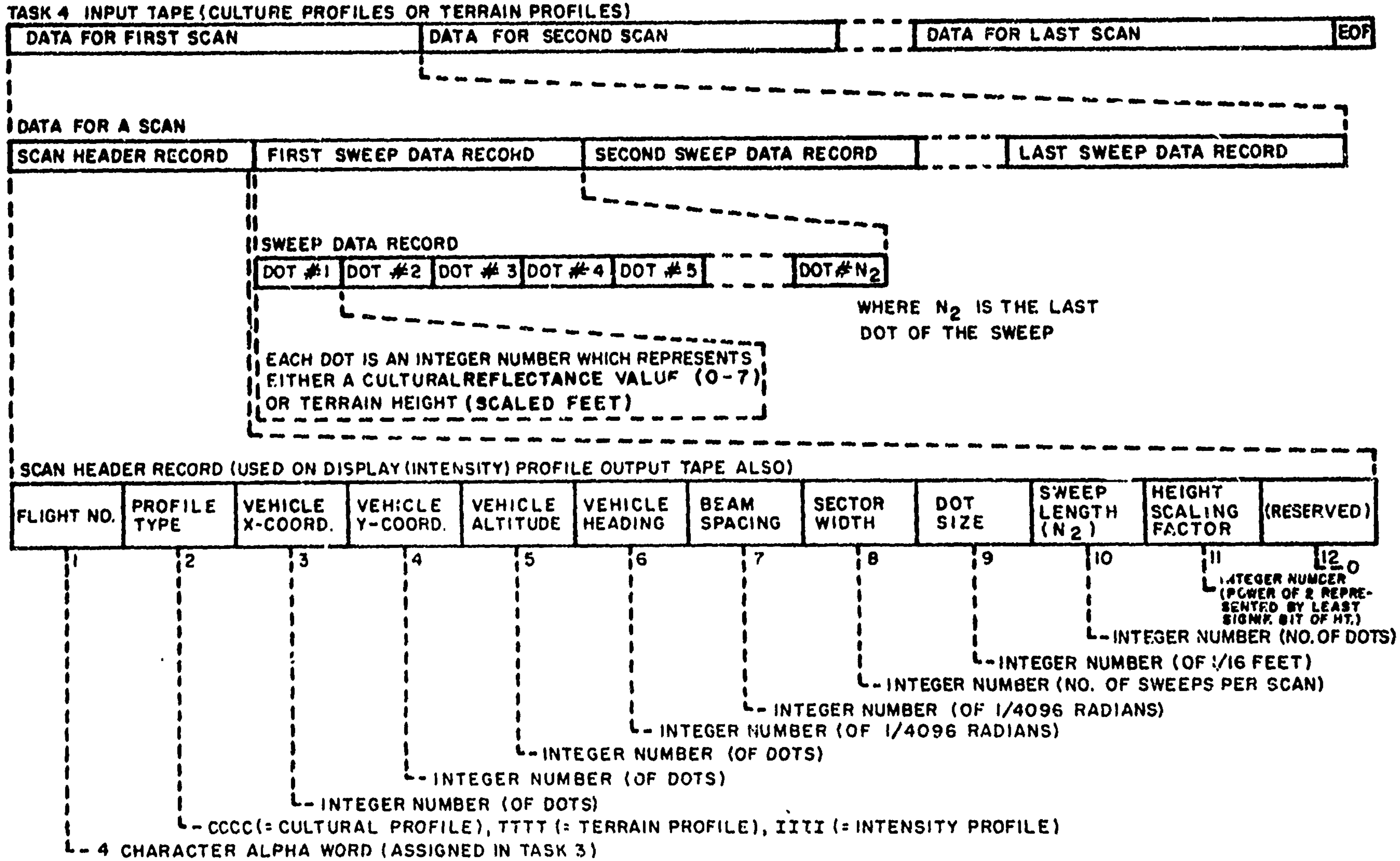


Figure 36 -- DI-3 INPUT TAPE FROM TASK 3

a tape file from Task 3, in accord with a user-specified table. The table is in the form of an 8 x 8 array whose row corresponds to the present reflectance level (0-7) and whose column corresponds to the value of the previous reflectance level; each entry in the array provides the value to be added to the current reflectance level. In general these entries will be zero but may take either algebraic sign, enabling increase or decrease in the apparent reflectance level at a boundary or transition (such as an increase at a water-land boundary or a decrease at a culture-terrain boundary) without affecting the reflectance level ascribed to the feature away from the boundary. As originally programmed no special effects are inserted; however, all current reflectance levels of "1", indicating water, are made negative. The water-negative convention is used by the following programs.

- CULTSCL Performs scaling of cultural reflectance to account for contrast and brightness settings and for logarithmic or other culture representation. (10) This program operates by looking up the output reflectance level corresponding to the input reflectance level in a user-specified table of eight entries. Negative numbers, indicating water, are left unchanged.

If the directions for processing the flight call for TERRAIN ONLY or BOTH FACTORS(11) then one sweep of the Terrain Profile tape from Task 3B is read and stored for processing by the read command. (12) Again, Figure 36 shows the format for the input tape(s) from Task 3(DI-3). If the option for CULTURE ONLY was requested for this flight, default values for terrain would be stored. (13) The default value for terrain is a uniform intensity of 0 for all dots in the sweep.

The sweep data given as terrain is processed by four subroutines which perform the following:

- SMOOTH Performs smoothing of the digitally generated profile to alleviate quantization and sampling effects and calculates smoothed terrain slope at each resolvable point. (14) This program unpacks and transforms the compacted terrain heights obtained from Task 3 as a tape file to floating point form.* It then invokes a sliding parabolic fit to the height profile with a

*This program, along with other Task 4 programs, accesses the scan header on the input file for scale factors and other parameters needed to perform the indicated functions.

user-specified number of points to be fit at each iteration. Specification of 3 points to be fit indicates no smoothing, since a parabolic fit to 3 points is exact; specification of 4 or more points to be fit indicates increasingly heavy smoothing on the terrain profile. Such smoothing is used to eliminate any kinks or gliches that might have occurred in the previous processing and enables a smoother terrain profile to be presented to the following computations.

o EARTH Compensates for earth curvature and atmospheric refraction by decreasing the height as the ground range increases, and calculates elevation angle of the ray to each resolvable point on the height profile for aircraft altitude being simulated.(15) The adjusted profile is the terrain height with respect to a plane perpendicular to the earth radius extended to the radar position. This program computes the adjusted height profile. The earth radius, normally set to 4/3 the geographic earth radius, is an assembly parameter which may be changed to account for different degrees of atmospheric refraction. A secondary output of this program is the slope of the line of sight between the radar and each point on the adjusted height profile.

o TERRFL Calculates terrain reflectance, occurrence of local shadow (at the backside of a hill and occurrence of remote shadow (from intervening hills) along the profile). (16) Terrain reflectance is based on Lambert's Law: the signal power level reflected by terrain is proportional to the cosine of the angle between the line-of-sight and the normal to the surface. The slope of the line of sight for each point is an input variable from the preceding program. The normal to the surface is determined by the central difference of the heights at adjacent points. Occurrence of local shadow determined by the angle between the line-of-sight to the terrain and the normal to the surface going from acute to obtuse, and remote shadow is determined by comparing the slope of the line-of-sight to each point with the most nearly horizontal slope previously obtained in tracing out the profile. Output of the program is a terrain reflectance profile, or return signal power for each resolvable increment of range, due to terrain alone. Any negative value indicates shadow; the shadow-negative convention is used by the following programs.

o TERSCL Performs scaling of terrain reflectance to account for contrast and brightness settings and for logarithmic or other terrain representations. (18) This program accesses a user-specified table of 16 entries; the table represents a piecewise linear approximation to the logarithm (or other desired) curve for converting signal power to a decibel formulation. Output is an intensity profile due to terrain in a form compatible with the intensity profile from culture and hydrography (levels 0-7).

Before continuing with the description of the process, a few notes about the input tapes need be given:

1. Where both terrain and cultural data are provided for a flight, the sequence and quantity of data must correspond as they were produced from the same flight path and display format controls.

   a. Each scan on the culture tape must have a corresponding scan on the terrain tape (vice-versa) for the same flight.

   b. The scans must be in the same order and, within a flight, must have the same number of sweeps per scan.

   c. Each sweep within a flight must have the same number of dots per sweep.

2. More than one flight may be present on either or both types of input tapes.

3. There may be more than one input tape for each type of data.

4. Sense Switch #2 "ON" indicates the current culture input tape(s) is(are) the last tape(s) of that type to be processed. Sense switch #3 "ON" does the same for terrain tapes. (At the beginning of each computer run these sense switches are automatically assumed to be "OFF".

   A single end-of-file signals the end of a tape.

Continuing with the description of processing, at this point one sweep of data is stored for terrain and one for culture (whether from tape or by default). If these sweeps are the first of the scan from the input tape(s) (19) a scan header is prepared and written to the output tape so the later sweep data will have descriptive information (20). If the sweep is not the first of a scan a header would have already been written so it is unnecessary to write another.

The following four programs combine the cultural and terrain data and insert effects on that combined sweep data:

o COMBIN Combines the two profiles, taking account of shadowing and superposition of ground features.(21) For each resolution element along the profile the cultural intensity is used, unless it references terrain, in which case the terrain intensity is used. However, if culture intensity is negative, indicating hydrography, or if terrain intensity is negative, indicating shadow, a zero intensity output is produced. The result of this program is the combined intensity profile in ground range coordinates.

o ATTEN Performs signal attenuation in accord with the antenna elevation pattern, and inserts random noise in the receiver.(22) This program accesses a user-specified table of thirty values, corresponding to the simulated antenna elevation pattern. That is, the slope of the line of sight is used to access the table of attenuations (given in 3-degree increments), and the attenuations are applied to the intensity profile. A simulated fan beam uses zero attenuation at each point. A user-prepared program module can be incorporated here to insert random noise: glint noise is inserted prior to antenna elevation pattern attenuation, and radar system noise is inserted afterward, if desired in the simulation.

o SRCONV Determines the slant range that corresponds to each resolvable point of ground range, associates points having the same slant range, and inserts the altitude hole.(23) This program effectively projects ground range resolution elements onto slant range resolution elements of the same size and performs an averaging process to distribute the intensities according to the fraction of each ground range bin that corresponds to each slant range bin. A filling-in

process is also performed, so that all slant range bins are accounted for, including those within the altitude hole (which are given zero intensity). This program is not called if ground range display format is directed.

o RNGEFF Simulates range-dependent effects such as time-varying gain and pulse stretching.(24) Provision for this program enables insertion of a user-prepared module which manipulates resolution elements along each intensity profile to incorporate such effects.

Certain effects and insertions must be made which involve examining data on neighboring sweeps. The Task 4 program will gather the necessary sweeps before writing the results to the output tape. But, in all cases, the number of sweeps (or pairs of sweeps if processing BOTH FACTORS) read from the input tapes.

The following effects and processes are performed on the combined sweep:

o AZEFF Simulates azimuth-dependent effects such as beam-spreading and side lobe phenomena.(25) Provision for this program enables the insertion of a user-prepared module which manipulates resolution elements across adjacent intensity profiles to incorporate such effects.

o GRAT Superposes a graticule (range rings and azimuth lines) on the display.(26) Ranges at a quarter, half, and three quarters of the maximum range, and an azimuth line corresponding to the aircraft centerline, are superposed on the displayable profiles. The intensity of this graticule is an input parameter and may be set to any desired value, lighter or darker than the average displayed intensity; it may also be turned off completely.

o RUNL Converts intensities at successive resolvable points to run-length code so that a sequence of identical intensities is compressed into a single display command.(27) These display commands, consisting of intensity and length of each run as shown in Figure 37, are arranged two to a word in the format required by the special display hardware. This program also counts the number of points and the number of runs per sweep for housekeeping purposes

and places the data describing the content of each profile in the output which will be written on tape for Task 5.

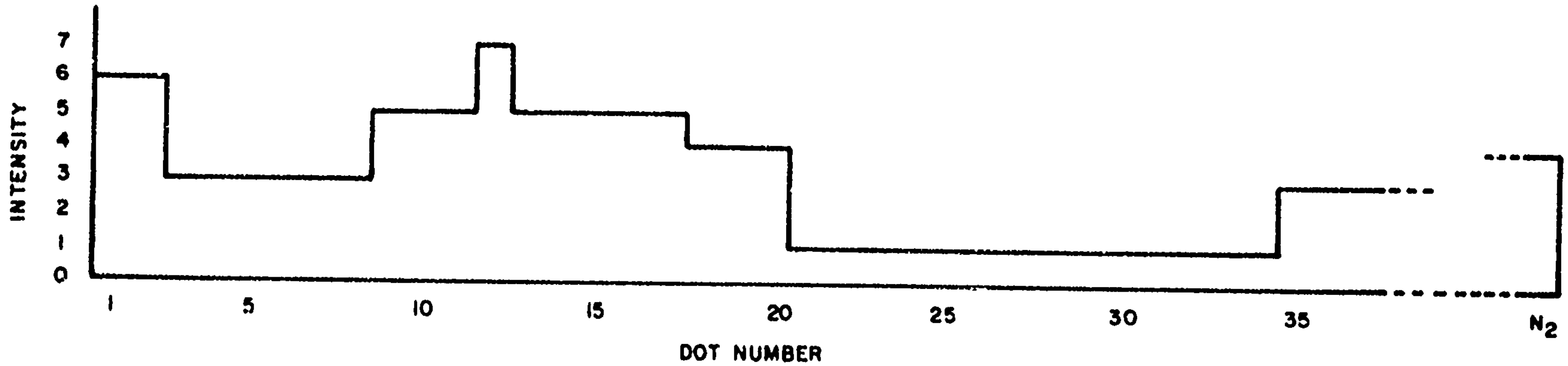
o ROTATE Develops the angle, of each sweep in display coordinates (relative to aircraft heading being simulated) for the display format implemented.(28) This program produces the header for each sweep which precedes the sequence of run intensities and length, independent of the actual content of each sweep profile and inserts this header into the output for Task 5. Each sweep within the scan may thus be processed by the display independently, since it comprises both a header describing its location, orientation, and run count, as well as intensities and length of each run along the sweep thus described.

The fully processed sweep data is now written to the Display (Intensity) Profile tape to be used later in Task 5.(29) The format for this output tape is given in Figure 38.(DI-4).

If more sweeps are to be read to complete the current flight being processed the program returns to the tape reading section. If a new flight is to be read the program starts from the beginning where new directions for processing are read. If there is no more input data, the job stops and Task 4 is complete.(30)

When a new scan is read, that scan will or will not be printed in the log as soon as it is read according to the STATUS options BY SCAN or BY FLIGHT.(31).

Figure 40 shows a schematic of the profile processing as extracted from the overall Task 4 process.



PLOT OF INTENSITIES OF DOTS

RUN NO.	INTENSITY	LENGTH
1	6	2
2	3	6
3	5	3
4	7	1
5	5	5
6	4	3
7	1	14
8	3	7
⋮		
N_3	4	24

LIST OF RUN-LENGTH DATA

INTENSITY LEVEL 0 ON THE DISPLAY IS "OFF"
INTENSITY LEVEL 7 ON THE DISPLAY IS THE BRIGHTEST

Figure 37 -- RUN LENGTH CODES

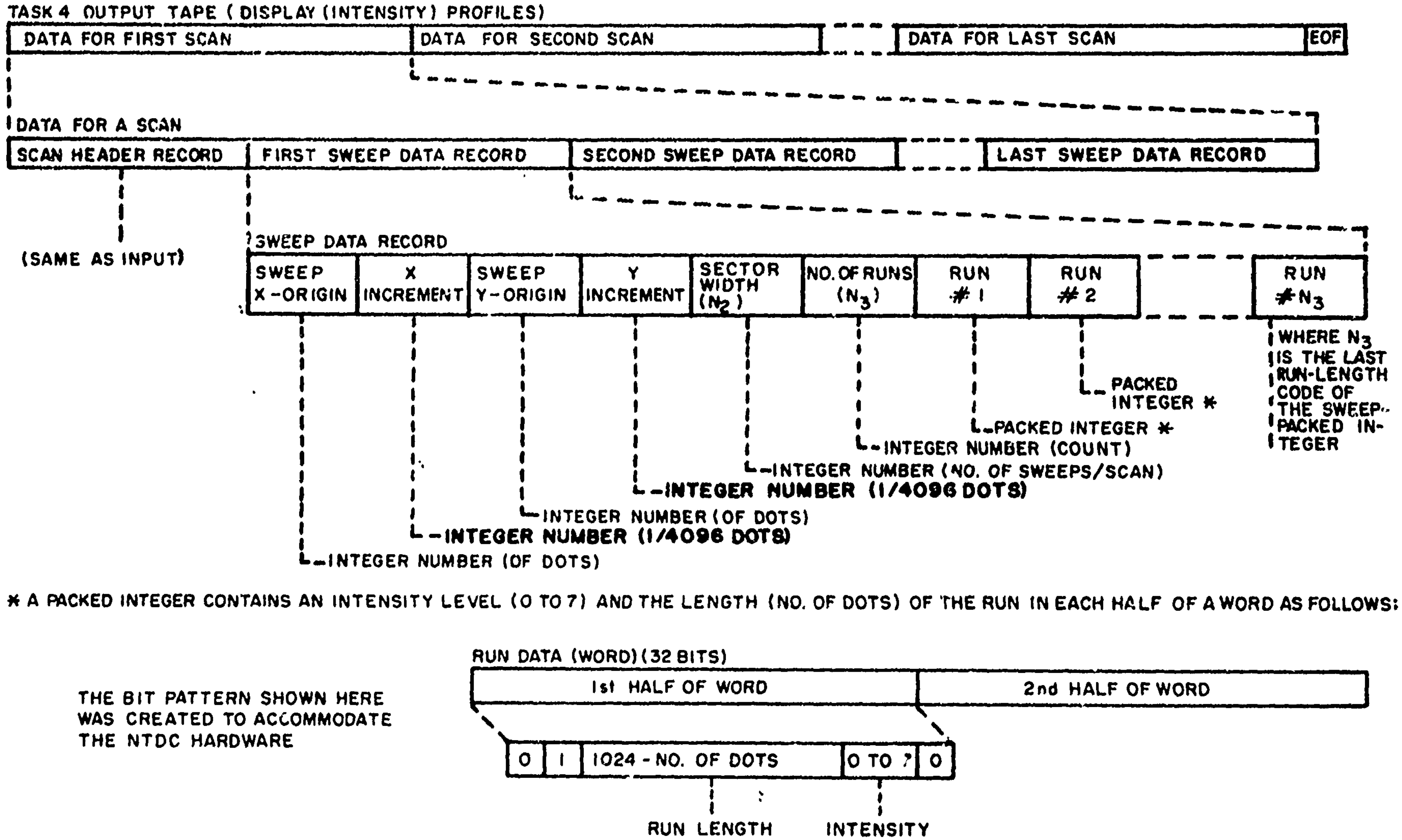


Figure 38 -- DI-4 OUTPUT TAPE FROM TASK 4

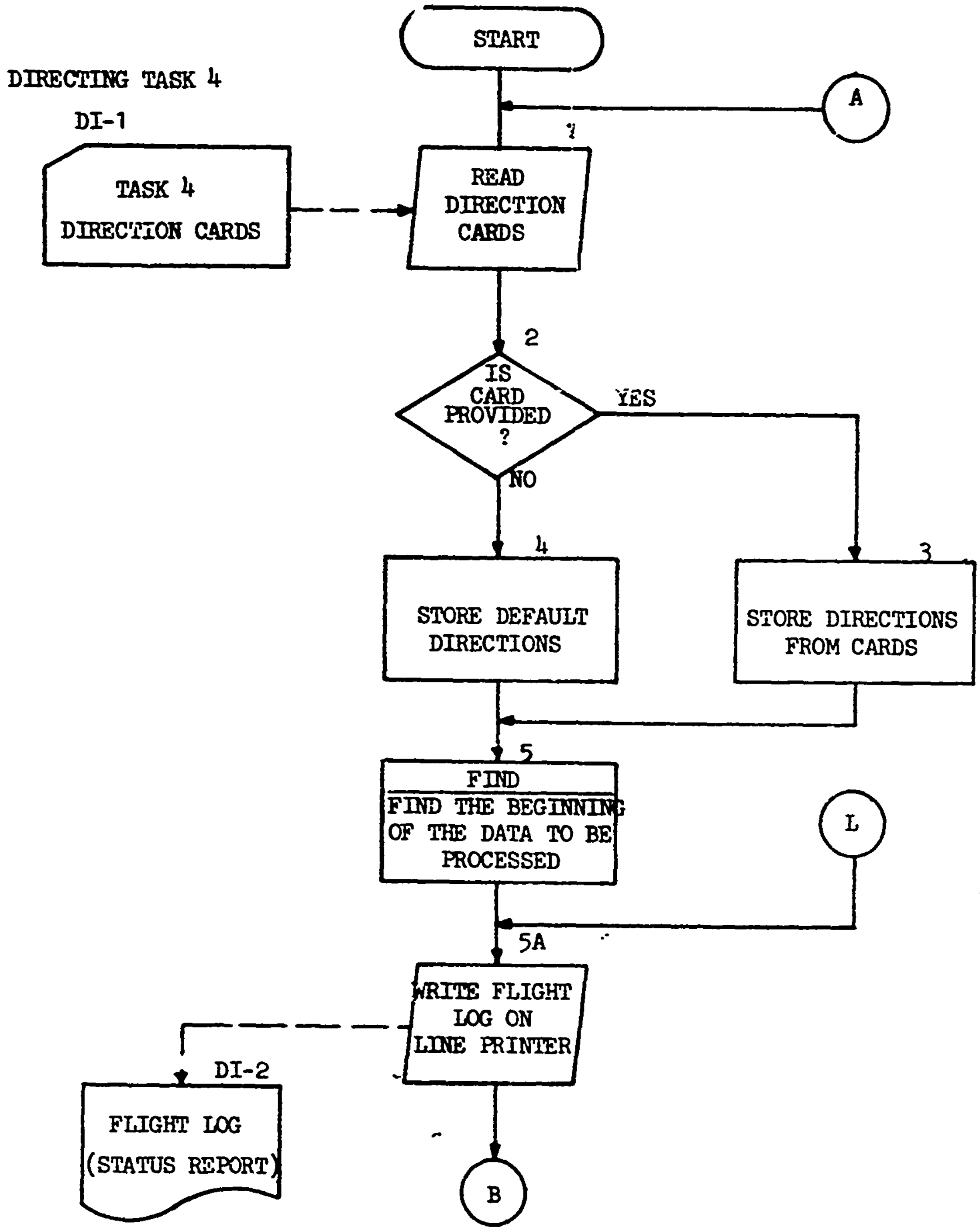


Figure 39 -- MACROFLOWCHART OF TASK 4 PROCESSOR
(Sheet 1 of 5)

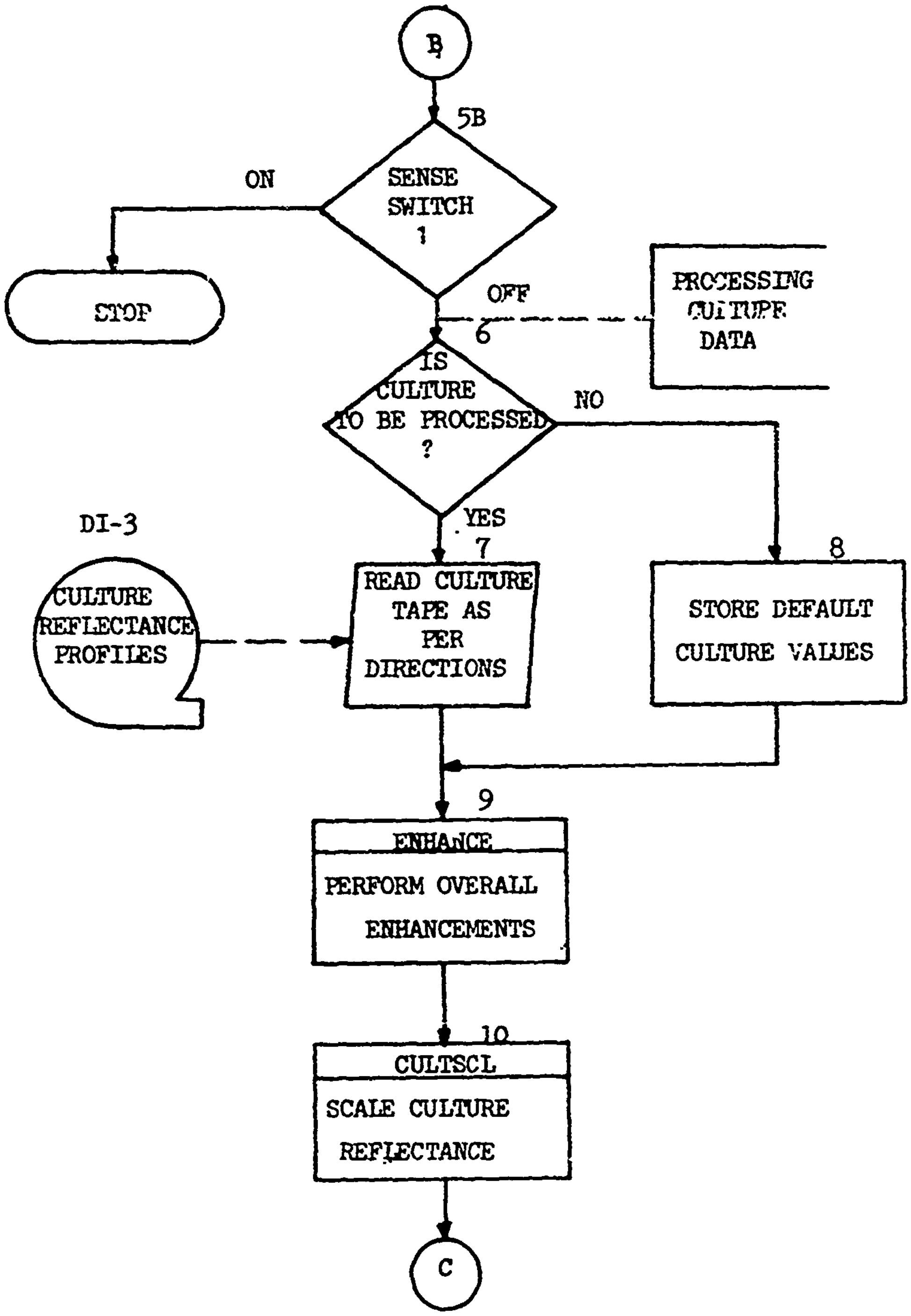


Figure 39 -- MACROFLOWCHART OF TASK 4 PROCESSOR
(Sheet 2 of 5)

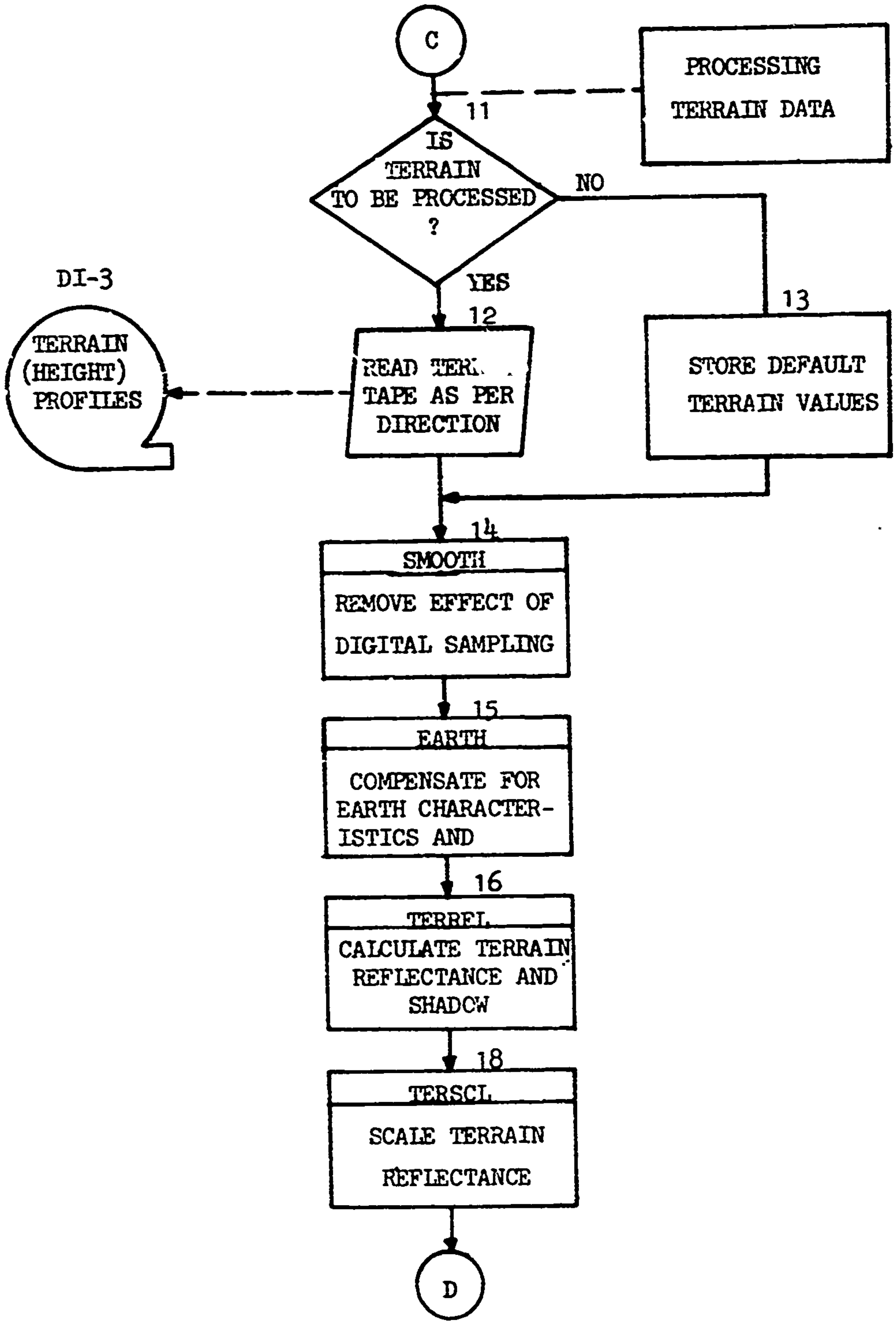


Figure 39 -- MACROFLOWCHART OF TASK 4 PROCESSOR
(Sheet 3 of 5)

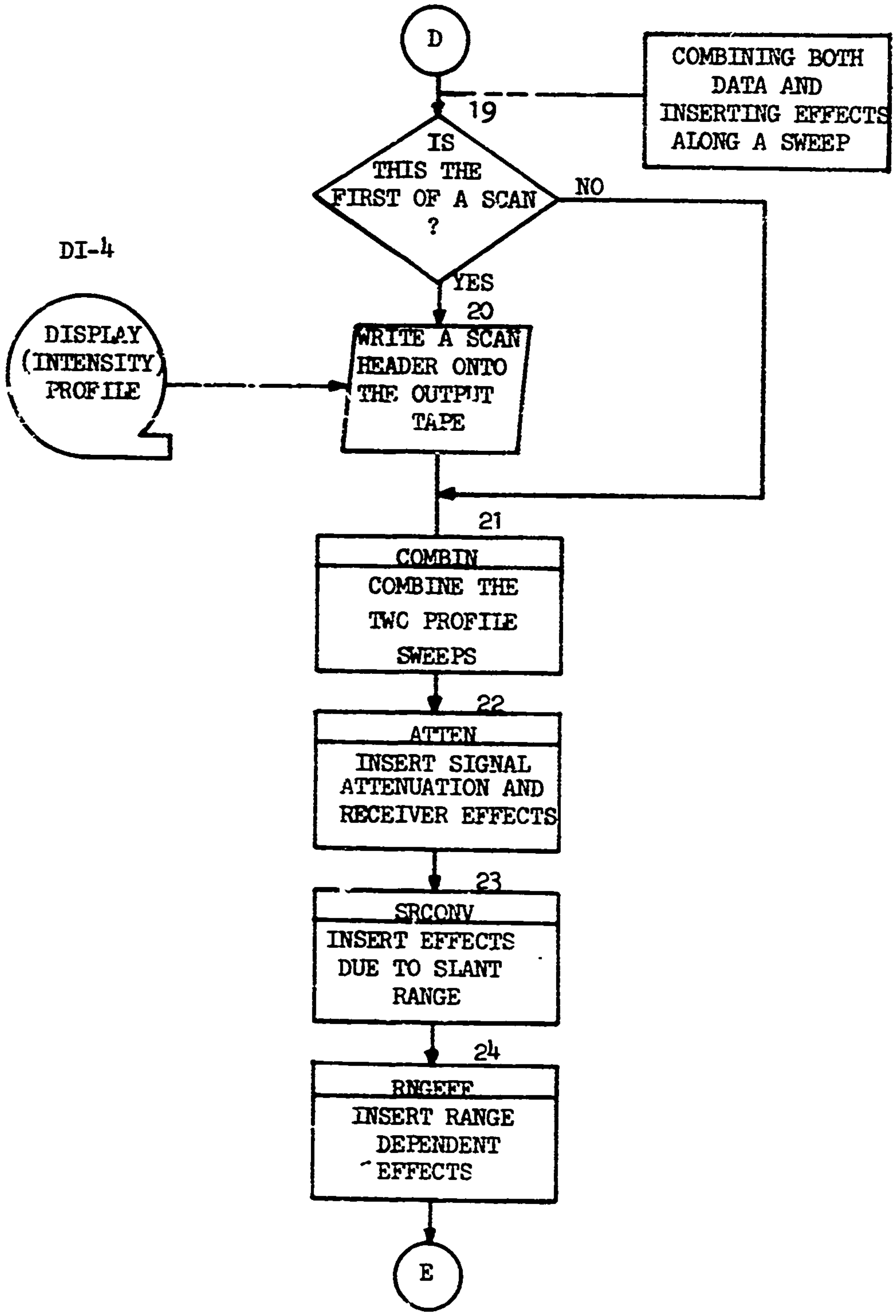


Figure 39 -- MACROFLOWCHART OF TASK 4 PROCESSOR
(Sheet 4 of 5)

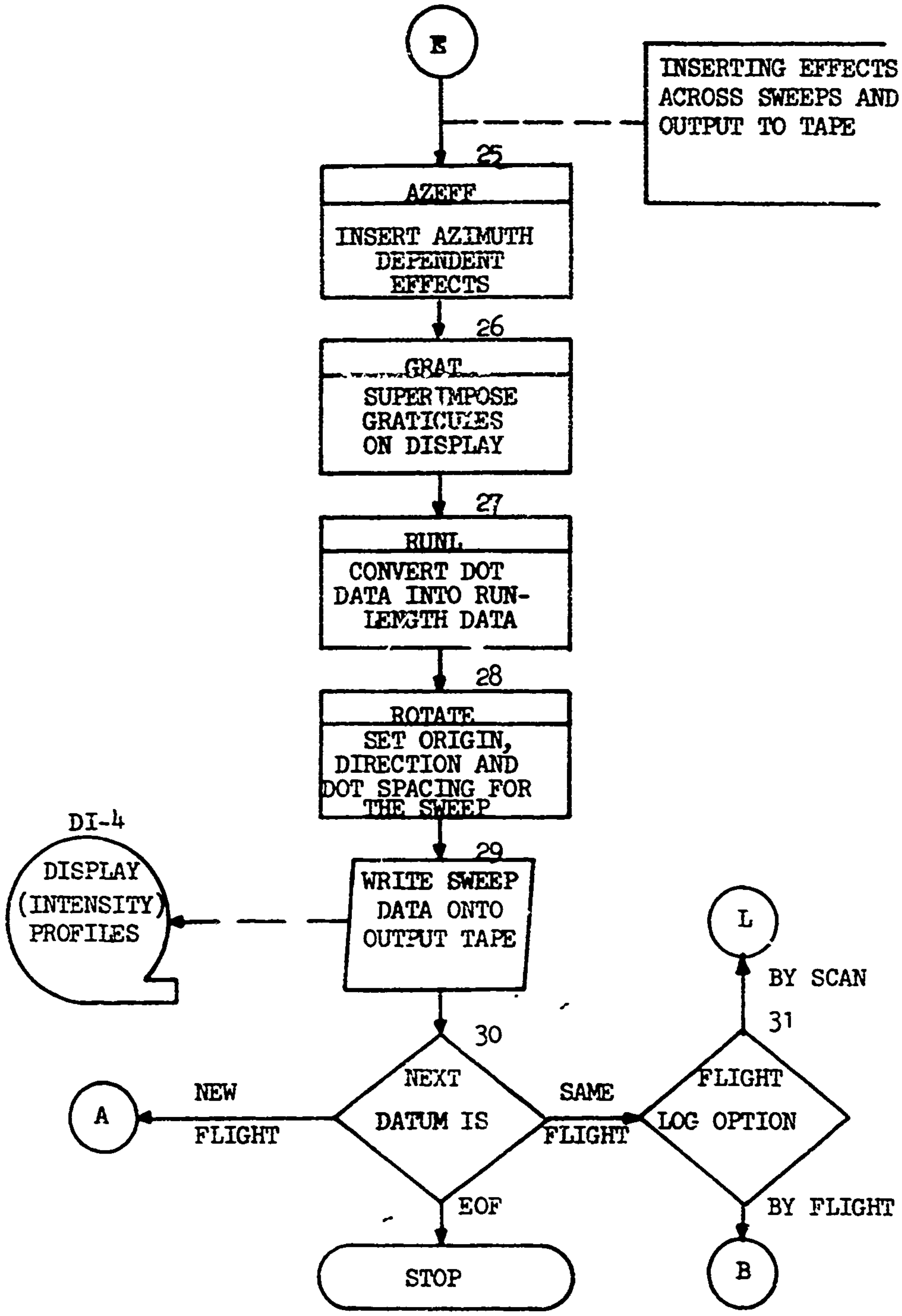


Figure 39 -- MACROFLOWCHART OF TASK 4 PROCESSOR
(Sheet 5 of 5)

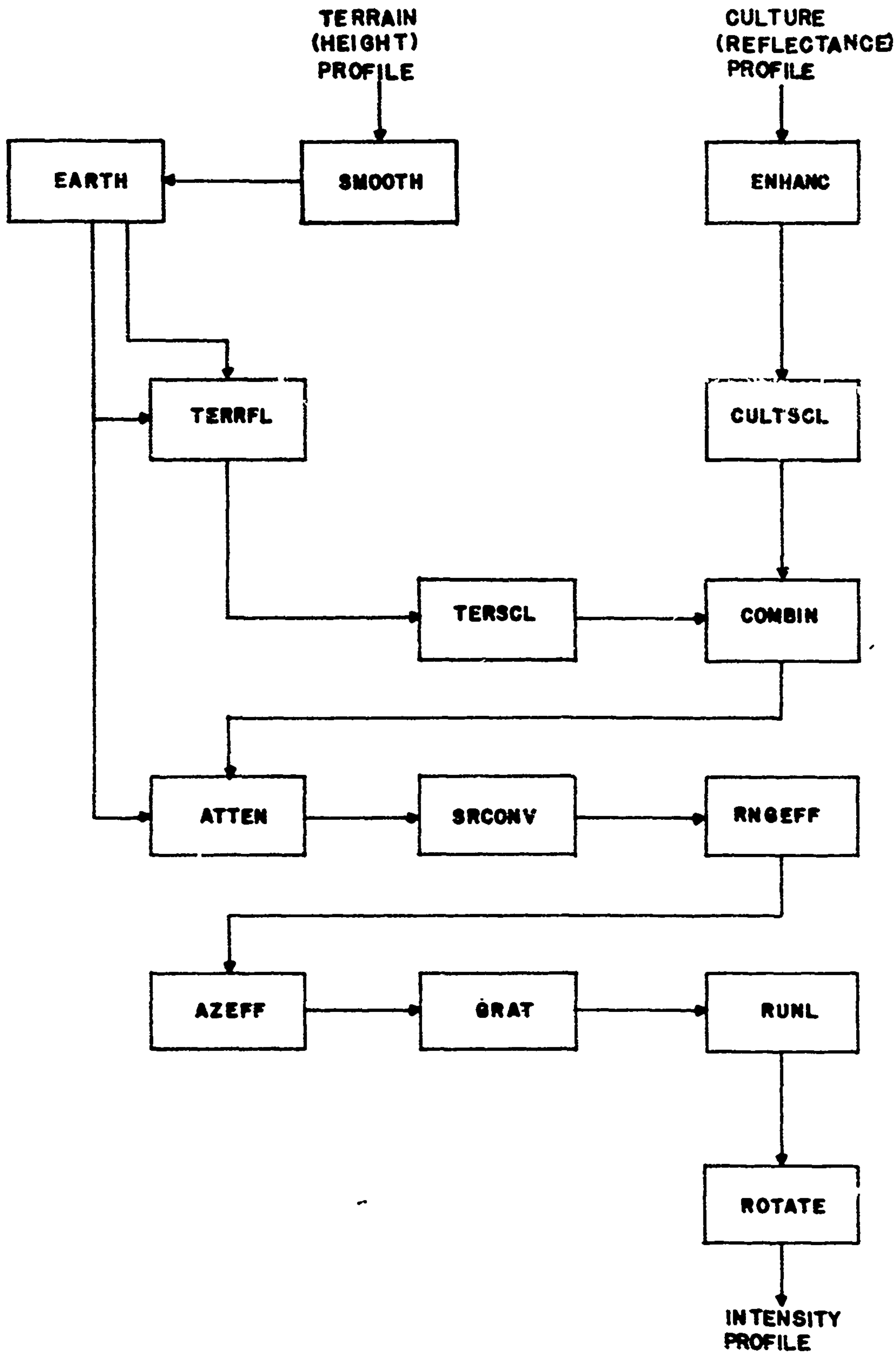


Figure 40 -- SCHEMATIC OF TASK 4 PROFILE PROCESSING

5.2 A SAMPLE TASK 4 JOB. This section contains a sample of the operation of a Task 4 computer run. The sample is not necessarily indicative of a typical run rather it attempts to show some of the variations which may be done with Task 4.

5.2.1. <u>Preparing a Task 4 Job</u>.

Input Tapes:

Culture Data (from Task 3A) -- on 3 tapes containing flights 14F3, 27A2, Ø71, 12G5, and 17G2.

Terrain Data (from Task 3B) -- on 4 tapes containing flight 30H5, 14F3, 27A2, 19B9, Ø7A1, and 12G5.

Display Tape to Contain:

1. Flight 14F3 is to contain information for both culture and terrain. Graticule light to be full bright with no smoothing factor on terrain. Status reports to be given and each scan processed.

2. Flight 27A2 is to contain terrain information only. Graticule light to be full bright with smoothing factor of 10. Status reports to be given with each scan.

3. Flight 27A2 is to contain culture information only. Graticule light to be full bright. Status reports to be given with each scan.

4. Flight Ø7A1 to contain both culture and terrain information. Graticule light to be full dark with no terrain smoothing. Status reports to be given only once for the entire flight. Ground Range view is desired.

5. Flight 17G2 to contain culture information only. Graticule light to be turned off. Status reports to be given with each scan.

The Task 4 Direction Cards to be loaded with the operating deck should be prepared as follows to yield the five displays required:

	COL 1	COL 13
1)	STATUS	BY SCAN
	GRATICULE	7
	SMOOTH	0
	14F3	BOTH FACTORS
2)	SMOOTH	10
	27A2	TERRAIN ONLY
3)	27A2	CULTURE ONLY
4)	GRATICULE	0
	SMOOTH	0
	STATUS	BY FLIGHT
	GROUND RANGE	
	∅7A1	BOTH FACTORS
5)	GRATICULE	-1
	STATUS	BY SCAN
	17G2	CULTURE ONLY

NOTE: Modifying option cards for a flight always precede flight processing card. Once the option is set it remains the same for each succeeding flight unless changed except for GROUND RANGE option which acts only on the flight number which it precedes.

The complete read to load, operating deck for a Task 4 run is assembled as follows:

```
!JOB.........
!ASSIGN F:10,(DEVICE,9T),(INSN,CULT),(TRIES,25)
!ASSIGN F:11,(DEVICE,9T),(INSN,TERR),(TRIES,25)
!ASSIGN F:12,(DEVICE,9T),(OUTIN),(OUTSN,DSPY),(TRIES,25)
!LOAD(GO),(UNSAT,(F4L1B)),(BI)
(put all program binary decks here)
!EOD
!BCD
!RUN
!DATA
(put all processing direction cards here)
!EOD
!FIN
```

As shown above, all culture tapes to be read (ring out) will be mounted as requested in the course of the run on the tape drive designated as logical unit 10. Terrain tapes will be mounted on logical unit 11 (ring out). Display (Task 4 output) tapes will be mounted on logical unit 12 (ring in).

At the start of the run all Sense Switches are assumed to be "OFF." Sense Switch 1 will be turned "ON" only if it is desired to stop the job in the middle of the run. There is no recovery from such a stop, however, the Display Tape will be complete to that point.

5.2.2 Procedure for Running a Task 4 Job. With the input and output tapes on hand and the operating deck (including the processing directions cards) in proper order, the deck is loaded into the card reader and the job started from the SIGMA-7-console. As the console requests that tapes be mounted, the operator mounts the first reel of culture, terrain or display tape on the appropriate transports. The job proceeds as indicated by tape movement and log printing.

During the course of the job, each time one of the three mounted tapes reaches an end-of-file mark, the tape will rewind and a message

will be typed on the console to mount the next tape containing like data. When the last culture tape is mounted Sense Switch 2 should be turned "ON" to indicate 'last culture tape'. When the last culture tape reaches end-of-file only request to process terrain data will be executed. When the last terrain tape is mounted Sense Switch 3 should be turned "ON" to indicate 'last terrain tape'. When the last terrain tape reaches end-of-file only requests to process culture data will be executed. If both types of input data are exhausted or the card reader shows no more flight processing directions the job stops. A message that the job is finished is typed on the console.

SECTION IV
DISCUSSIOl.

This section discusses the experience with the programs. Unfortunately time has not permitted much experience with the production programs. In fact, the Task 5 program has not yet been written so that the simulated radar pictures from Tasks 3A, 3B, and 4 have never been placed into pictorial form. It is hoped that by the time the final version of this report is published, pictures will have been made to demonstrate the flexibility of the system created under this contract.

Much experience was gained, however, in the operation of the data preparation programs: Tasks 1 and 2. These two tasks had one very striking thing in common: they both took over twenty-four hours of computer time on NTDC's SIGMA-7 computer. For production programs with which experimentation is to be performed, this is not a good characteristic. However, Task 1 is not designed as a production program, but rather it is a one-time tool to prepare a problem area of 60x270 miles as the data base for the experiments to be performed by the rest of the system. For experimental runs, it is not necessary to use the Task 2 processor for the whole data base, but rather a small sub-area is quite adequate. Processing time of the Task 2 program is directly proportional to size of the area processed.

The limited experience with Tasks 3A, 3B, and 4 indicates that while Tasks 3A and 4 have rapid computer processing times, Task 3B does not. This is because Task 3B was written in FORTRAN by NTDC direction. As it currently exists, Task 3B takes 45 seconds to produce one terrain height profile sweep of 777 resolution elements in length. This implies that the data for each radar scan requires over an hour of computer execution time (depending on the scan pattern to be simulated). An equally flexible, more simply constructed, and more comprehensable program could have been written in assembly language. Analysis indicates it would produce the same terrain height profile in about a tenth the time of the FORTRAN program.

The long execution times experienced are due to limitations imposed by the SIGMA-7's operating system and more specifically the FORTRAN (both compiler and run-time subroutine library) included in that operating system, as discussed below.

1. TIMING EXPERIENCE WITH TASK 1.

The Task 1 program stores a large array onto the RAD. The major bottleneck in processing is accessing this array. Each input data column must be added to several columns in the disc array. The structure of the RAD is such that the required updating may be done in two revolutions of the RAD for each column updated. Table 5 shows the number of columns to be updated for each map, the number of input columns, and the total revolutions required for the processing. The RAD rotates at 1774 revolutions per minute. From this the Task 1 program can be expected to take approximately 10 hours.

The above timing assumes that the programs which access the RAD operate with reasonable efficiency. Since the Task 1 program is found to require 2.5 times the predicted figure, this assumption is false. The FORTRAN random access feature is not an efficient program. An extremely sophisticated program to allow random accessing of the RAD via FORTRAN could be written that would result in a Task 1 processing time of 7.5 hours. A relatively simple program allowing the same facility could be written that would result in a Task 1 processing time of 10 to 12 hours. Neither of these programs, however, can use the BPM I/O facilities. After study of the BPM manual and searching for some random access facility within BPM, one is forced to conclude that the authors of the FORTRAN subroutine library deserve commendation to have supplied a random access facility at all.

MAP NUMBER	COLUMNS TO BE UPDATED	INPUT COLUMNS	REVOUTION TIMES
1	19	1,325	50,350
2	39	1,325	103,350
3	58	1,324	153,584
4	78	1,325	206,700
5	95	1,305	253,170
6	118	1,327	313,172
			1,080,326

÷ 1,774 RPM = 609 minutes

÷ 60 min./hour = 10 hours

TABLE 5 -- TASK 1 PROCESSING TIME PREDICTION

2. TIMING EXPERIENCE WITH TASK 2.

As originally written the Task 2 program would have taken 60 hours to process the total problem area. By hand-compiling one bottleneck subroutine, this time was reduced to 24 hours. An additional 25% reduction could have been obtained with the use of additional 1,500 words of memory and major rewriting of this routine. The 1,500 words of memory were not available, and the reprogramming time would have exceeded the processing time saved.

The 60 hour to 24 hour saving is all due to the hand-compilation of the subroutine COMP. There were two differences between the original machine-compiled version and the hand-compiled version: (a) the mathematical laws of commutation, distribution, and association were applied to the formula and loops to take advantage of the automatic indexing facilities of the SIGMA-7, and (b) the automatic indexing facilities were used to implement the index and DO-loop statements of the program. If only (a) were done and the result fed to the FORTRAN compiler, no saving in time would have been realized.

The FORTRAN compiler of the SIGMA-7 cannot remember that the needed index is still in a register and must reload it. Other FORTRAN compilers on other computers have this facility. In the case of the COMP subroutine this accounts for a 75% possible saving.

Independent of the DO statement, the FORTRAN compiler on the SIGMA-7 has a standard way of implementing DO-loops. Therefore it does not take advantage of the automatic indexing facilities in the computer. Other FORTRAN compilers do not allow as much variation in the DO-statement, and therefore can produce a better standard DO-loop implementation. The DO-loop inefficiencies account for the remaining saving.

3. TIMING EXPERIENCE WITH TASK 3B.

The Task 3B program takes 45 seconds to produce a terrain height profile of 777 resolution elements. The major culprit is the FORTRAN subroutine linkage. In order to allow the flexibility in word size required of the multrix simulator, for example, several subroutines were written to perform the intermediate operations. PRA originally planned to write METASYMBOL procedures instead of FORTRAN subroutines. The multrix simulation subroutine would have been generated by the METASYMBOL assembler with all the shifts and masking as constants. Thus only a few instructions would be executed to evaluate each term, as opposed to the several instructions required to call all the subroutines to evaluate the same term. A factor of 10 improvement is possible by the use of METASYMBOL procedures.

A detailed example is now presented. The FORTRAN coding for the "delimit" function, required several times in the multrix simulation for each resolvable point on the terrain, is:

```
SUBROUTINE DELIMIT(ISIG,ITAU,NSIG,NTAU,VALUE)
DATA MASK/Z7FFFFFFF/
DR = ITAU-NTAU
VALUE = ISA(VALUE,DR)
L = NSIG-NTAU+2
DELIMIT = ISA(ISL(VALUE,32-L),L-32)
RETURN; END
```

The term DELIMIT (10,-5,8,0,X) results in approximately 50 instructions being executed.

Writing the equivalent METASYMBOL procedure would result in the term DELIMIT (10,-5,8,0,X), from which the following code would then be generated:

```
LW,0     X     LOAD X
SAS,0   -5     Shift right 5 (i.e., by DR)
SLS,0   22     Shift left 22 (e.e., by 32-L)
SAS,0  -22     Shift right 22 (i.e., by L-32)
STW,0          Store result (i.e., return value of DELIMIT)
```

At compile time the number DR and L would be computed to be -5 and 10 respectively. It can be seen that only 5 instructions need be executed to perform the "delimit" function.

SECTION V

REFERENCES

1. Pennsylvania Research Associates Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 4 (U), U.S. Naval Training Device Center, Tech. Rpt. 68-C-0155-4, March 1969.

2. Pennsylvania Research Associates Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 2 (U), U.S. Naval Training Device Center, Tech. Rpt. 68-C-0155-2, November 1968.

3. Pennsylvania Research Associates Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report Nr. 3 (U), U.S. Naval Training Device Center, Tech. Rpt. 68-C-0155-3, January 1969.

4. Mosteller, W. S. and Wolfgang, Paul A. T., Development of a Hybrid Radar Landmass Simulator: Engineering Report 5 (U), U.S. Naval Training Device Center, Tech. Rpt. 69-C-0086-1, March 1970.

5. Pennsylvania Research Associates Inc., Incorporation of Terrain Data in Radar Landmass Simulator: Work Plan Report, U.S. Naval Training Device Center, Tech. Rpt. 70-C-0262-1, July 1970.

6. Pennsylvania Research Associates Inc., Development of a Hybrid Radar Landmass Simulator: Engineering Report 6 (U), U.S. Naval Training Device Center, Tech. Rpt. 69-C-0086-2, March 1970.

7. Forsythe, George E. and Moler, Clive B., Computer Solutions of Linear Algebraic Systems, Prentice-Hall Inc., pp. 68 -70 and 77-99, 1967.

8. Pennsylvania Research Associates Inc., Task 1 Program Documentation Contract N61339-70-C-0262, PRA Document U71-1293, June 1971.

9. Pennsylvania Research Associates Inc., Task 2 Program Documentation Contract N61339-70-C-0262, PRA Document U71-1294, June 1971.

10. Pennsylvania Research Associates Inc., Task 3A Program Documentation Contract N61339-70-C-0262, PRA Document U71-1295, June 1971.

11. Pennsylvania Research Associates Inc., Task 3B Program Documentation Contract N61339-70-C-0262, PRA Document U71-1296, June 1971.

12. Pennsylvania Research Associates Inc., Task4 Program Documentation Contract N61339-70-C-0262, PRA Document U71-1297, June 1971.

Naval Training Device Center, Orlando, Florida	DESCRIPTORS
TR NAVTRADEVCEN 70-C-0262-2 UNCLASSIFIED AD_________	Radar Landmass Simulator Culture Data Displays Terrain Data Displays Digital Computer Computer Programming Multiprogramming
DEVELOPMENT OF A HYBRID RADAR LANDMASS SIMULATOR: ENGINEERING REPORT 7(U). June, 1971, 136+iv pp, 40 illus., 5 tables, 12 refs.	Pennsylvania Research Associates Inc. Wolfgang, Paul A. T.
An integrated set of computer software is described for the generation of airborne radar displays, simulating the earth's topographic features and man-made objects to the extent they are apparent on the indicator of a modern scanning radar, as a function of the aspect from which the earth is viewed and the salient characteristics of the viewing radar system. The programs are based on previous PRA work in the simulation of air-to-ground radar displays and provide a versatile laboratory tool for the evaluation of radar simulation techniques with a variety of user options.	N61339-70-C-0262